Daniel Creangã

Licenciatura em Ciências e Engenharia Informática

Gestão da Infraestrutura no Datacenter (Monitorização / Gestão de Serviços IT / Automação)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Orientador: João Sanches, Senior Consultant, Unipartner IT Services, S.A.

Co-orientador: Paulo Orlando Reis Afonso Lopes, Assistant Professor, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri

Presidente: Doutor Pedro Abílio Duarte de Medeiros, FCT NOVA

Vogal: Doutor Hervé Miguel Cordeiro Paulino, FCT NOVA

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2019

Gestão da Infraestrutura no Datacenter (Monitorização / Gestão de Serviços IT / Automação)

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2]. [1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Quero dedicar a minha família este esforço,
esforço que não é comparável aos sacrifícios que vocês fizeram
de forma a ser possível chegar a este momento...
Um especial obrigado a todos que me acompanharam
e que continuam a fazer parte do meu crescimento.

Agradecimentos

Gostava de agradecer à *Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa*, com especial relevo aos membros do *Departamento de Informática* que tornaram esta dissertação de mestrado possível, assim como aos professores que se empenham todos os dias para transmitir e criar conhecimento nesta área. Um especial obrigado ao *Professor Paulo Lopes* pela ajuda com a correção e aconselhamento durante a dissertação de mestrado.

Em segundo lugar gostava de agradecer à empresa *Unipartner* que me acolheu, proporcionou os meios e incentivou-me a apreender e investigar em conformidade com as diferentes realidades empresariais existentes em Portugal nesta área. Não obstante, durante o percurso da dissertação e crescimento pessoal, fui apreendendo, penosamente, à custa de alguns erros, com um modelo de referência, o meu orientador *João Sanches*, que me permitiu crescer e desenvolver habilidades para lidar com os diferentes desafios distribuídos pelos variados projetos. Gostava de agradecer aos meus colegas de equipa, nomeadamente *Tiago Fernandes* por alimentar a minha curiosidade, pelo apoio contínuo e aconselhamento.

Por último, agradeço à minha família e amigos pelo constante apoio e equilíbrio durante este percurso académico.

Resumo

O sucesso da atividade empresarial depende da capacidade de adaptação à mudança. Isto é especialmente verdade nas empresas no ramo tecnológico, em que o ritmo de inovação é superior às outras áreas. A capacidade de uma organização competir à escala global é um desafio a ter em conta devido a heterogeneidade das necessidades e características dos clientes: língua, legislação, modo de pagamento, grau de desenvolvimento das infraestruturas de comunicação e transporte, características do produto.

Numa organização a responsabilidade de adaptação à mudança torna-se um desafio que recai maioritariamente sobre uma sub-unidade da organização, o departamento de IT, força impulsora responsável por suportar os processos de negócio e transformação dos produtos ou serviços, sendo fulcral uma correta gestão da infraestrutura IT, dos roles (papéis) assinados aos profissionais de IT e uma visão completa do estado da infraestrutura.

Independentemente da empresa ou área de atuação, o volume de dados gerado apresenta desafios quanto ao seu armazenamento, tratamento e interpretação. As organizações reconhecem a importância e o possível retorno no investimento na sua unidade IT de acordo com as suas necessidades estratégicas; contudo, este investimento em recursos computacionais não foge aos desafios operacionais e de gestão que permitem tirar melhor partido destes mesmos recursos. É expectável a carga sobre as organizações IT continuar a aumentar e consequentemente a gestão IT tornar-se mais complexa e mais automatizada, necessitando de uma equipa IT mais especializada.

Para ter controlo sobre a infraestrutura IT necessitamos de ferramentas de monitorização e gestão que nos permitam ter uma visão consolidada e orquestrada sobre o "estado de saúde"da infraestrutura e dos serviços e aplicações suportados.

Gerir e otimizar infraestruturas é um processo contínuo e evolutivo que, para além das estratégias de melhoria do produto/serviço, requer também estratégias de automação e recuperação em caso de falha de serviços para um conjunto de eventos. Assim sendo, este trabalho pretende mostrar as potencialidades de uma solução de gestão centralizada de infraestruturas, aplicações e serviços, com o objetivo de reduzir a complexidade de tarefas, bem como os custos e riscos para as organizações.

Palavras-chave: Gestão IT, Flexibilidade, Monitorização, Automação, Virtualização.

Abstract

In an organization, sustaining business growth depends on the ability to successfully adapt/respond to new trends/challenges.

This is markedly visible in IT organisations, where the rhythm of change is superior to other companies. For instance, the ability of an enterprise to compete on a global scale requires attention to the heterogeneity of client needs and characteristics, such as language, culture, payment methods, level of infrastructure development, legislation, etc.. All these factors must be included in the IT business process.

In an organization the responsibility to adapt to change is assigned to the IT department, which must support business processes and transformation of products or services. It was never so important to implement a correct management of the IT infrastructure, of the roles assigned to the IT professionals and have a global vision of the infrastructure health status.

The volume of data generated in any organization presents challenges to storage, filtering and processing the information. Enterprises recognize the importance and value returned by the investments on their IT units. However, this investment must be complemented with a correct management to maximize the utilization of these resources. It is likely the amount of work will continue to grow and, consequently, IT management will become more complex and will need automation, requiring a more specialized IT team.

In order to have control acontrol of the IT infrastructure we need monitoring tools to provide an overview of the health status of our infrastructure, avoiding being drowned with de-centralized or non-organized information, procedures or tools. Planning and building and infrastructure is a continuous task, encompassing not only on strategies to improve products or services but also to automate and recover with minimum impact in case of failures.

Keywords: IT Management, Flexibility, Monitorization, Automation, Virtualisation.

Índice

Lista de Figuras

Lista de Tabelas

Listagens

Glossário

.NET	*".NET is the brand name for a set of proprietary Microsoft frameworks and technologies founded on XML web services standards"* [1].
Bare-metal	Serve para descrever ambientes de IT em que o sistema operativo é instalado diretamente no *hardware*, em vez de ter uma camada de *software*, por exemplo, o *Microsoft Hyper-V* que é um sistema para gerir ambientes virtualizados.
Deployment	É a tarefa de instalar um *software* em diversas estações de maneira simples e eficiente com objetivo de organizar, facilitar e agilizar a manutenção da rede local após a sua implementação.
Failover	O processo de *failover* diz respeito a falha de componente de um sistema, e o sistema conseguir continuar a fornecer um nível de serviço mínimo, passar a carga de trabalho para outro servidor.
Híbrido	Um ambiente IT híbrido refere-se a combinação de *datacenters* internos ou locais com nuvens públicas ou privadas.
Log	É o termo utilizado para descrever o processo de registo de eventos relevantes num sistema computacional. Um ficheiro de *log* pode ser utilizado para monitorização e diagnóstico de problemas.
Multi-tenant model	*Multi-tenancy* é uma arquitetura em que uma única instância de aplicação *software* serve múltiplos clientes. Cada cliente é um *tenant*. Em *Cloud Computing* é possível devido a conceitos como a virtualização e controlo remoto.

Node	Na área de Informática, uma árvore é uma estrutura de dados que começa pela raiz e se estende para vários ramos e sub-ramos. Cada ramo interliga dois nós ou *nodes*.
On-premises	Uma infraestrutura diz-se *on-premises* quando está situada nas instalações físicas de uma organização.
Patch	É código adicionado sobre o *software* base com objetivo de melhorar a usabilidade ou performance do mesmo. São importantes na aplicação da correção de vulnerabilidades de segurança.
Python	É uma linguagem de programação interpretada de alto nível, multi-paradigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedimental.
Report Builder	É uma ferramenta ou um *software* que a base dos resultados das *queries* Strucured Query Language (SQL) sobre uma base de dados constrói relatórios personalizados.
Ruby	Ruby é uma linguagem de programação interpretada multi-paradigma, de tipagem dinâmica, com gestão automática de memória. *Ruby* suporta programação funcional, orientada a objetos, imperativa e reflexiva.
Script	São um conjunto de instruções computacionais dentro de um programa que é interpretado em tempo de execução, daí ser uma ferramenta muito versátil e potencialmente perigosa quando usada para fins maliciosos.
Standalone	Normalmente um servidor que executa autonomamente e não faz parte de um grupo.

Siglas

ACS	Audit Collection Service.
AD	Active Directory.
ADR	Automatic Deployment Rule.
API	Application Programming Interface.
ARM	Azure Resource Manager.
AWS	Amazon Web Services.
BITS	Background Intelligent Transfer Service.
BPaaS	Business Process as a Service.
CAS	Central Administration Site.
CLI	Command-Line Interface.
CLR	Common Language Runtime.
CMDB	Configuration Management Database.
CMS	Configuration Management System.
CPU	Central Process Unit.
CSV	Comma-Separated Values.
DB	DataBase.
DC	Domain Controller.
DHCP	Dynamic Host Configuration Protocol.
DP	Distribution Point.
DPM	Microsoft System Center Data Protection Manager.
DR	Disaster Recovery.
DSC	Desired State Configuration.
DSL	Domain Specific Language.
EC2	Elastic Compute Cloud (AWS).

FQDN	Fully Qualified Domain Name.
GB	GigaByte.
GUI	Graphical User Interface.
HTML	Hyper Text Markup Language.
IaaS	Infrastructure as a Service.
IaC	Infrastructure as Code.
IBM	International Business Machines Corporation.
IP	Integration Pack.
IP	Internet Protocol.
ISS	Internet Information Services.
IT	Information Technology.
ITIL	Information Technology Infrastructure Library.
ITSM	Information Technology Service Management.
JSON	JavaScript Object Notation.
KVM	Kernel-based Virtual Machine.
LCM	Local Configuration Manager.
LDAP	Lightweight Directory Access Protocol.
MMA	Microsoft Monitoring Agent.
MOF	Managed Object Format.
MP	Management Pack.
NASA	National Aeronautics and Space Administration.
NIC	Network Interface Card.
NIST	National Institute of Standards and Technology.
NT	New Technology.
ODBC	Open Database Connectivity.
OLE	Object Linking and Embedding.
OS	Operating System.
OU	Organizational Unit.

PaaS	Platform as a Service.
PC	Personal Computers.
RAM	Random Access Memory.
REST	Representational State Transfer.
SaaS	Software as a Service.
SCCM	Microsoft System Center Configuration Manager.
SCEP	Microsoft System Center Endpoint Protection.
SCOM	Microsoft System Center Operations Manager.
SCORCH	Microsoft System Center Orchestrator.
SCSM	Microsoft System Center Service Manager.
SCVMM	Microsoft System Center Virtual Machine Manager.
SLA	Service Level Agreement.
SMS	Systems Management Server.
SNMP	Simple Network Management Protocol.
SPN	Service Principal Name.
SQL	Strucured Query Language.
SSH	Secure Shell.
SSL	Secure Sockets Layer.
TCP	Transmission Control Protocol.
TMG	Microsoft Forefront Threat Management Gateway.
UDP	User Datagram Protocol.
UI	User Interface.
vCPU	virtual Central Process Unit.
VM	Virtual Machine.
VMM	Virtual Machine Monitor.
vNIC	virtual Network Interface Card.
vRAM	virtual Random Access Memory.
WDS	Windows Discovery Services.
WMI	Windows Management Instrumentation.
WSUS	Windows Server Update Services.

XML Extensible Markup Language.

Capítulo 1

Introdução

1.1 Contexto

O contexto desta dissertação é o das tecnologias de gestão e aprovisionamento de uma infraestrutura virtualizada maioritariamente composta por *Windows Servers* em ambiente Híbrido, i.e., no qual as funcionalidades disponibilizadas *on-site* podem estender-se para ambientes *cloud* e vice-versa.

A dissertação realiza-se em contexto empresarial, numa empresa líder no sector de tecnologias de informação, a *Unipartner IT Services, S.A.* Esta é parceira directa da *Microsoft Portugal* e abrange um vasto leque de especializações, sendo de destacar as que se referem a tecnologias *Microsoft* – 16 competências, das quais 15 são *Gold*. Assim, foi-me dada a oportunidade de conhecer o panorama tecnológico de alguns dos seus clientes e recolher informações, *in loco*, de como funciona a dinâmica tecnológica da qual as empresas dependem para dar continuidade ao negócio e prosperar.

1.2 Motivação/Objetivos

À medida que os negócios evoluem, as empresas sentem a necessidade de sistemas informáticos que aumentem a eficiência e produtividade dos desafios operacionais e de gestão. Existe um conjunto de tarefas de manutenção periódicas (diárias, semanais, mensais, semestrais ou anuais) que regulam o bom funcionamento da infraestrutura. Cabe às equipas de IT garantir uma infraestrutura saudável, o que naturalmente consome mão-de-obra e tempo que pode ser melhor aproveitado se as tarefas forem agilizadas com ajuda de ferramentas desenvolvidas para este propósito – ferramentas que foram concebidas para alertar para potenciais problemas, gerir os serviços IT e automatizar alguns componentes/procedimentos.

A orquestração (execução coordenada) de várias tarefas computacionais na infraestrutura consegue-se partindo da automação de pequenos sub-problemas sob a forma de *scripts* e *queries*; naturalmente, tal não dispensa analisar o custo/benefício de desenvolver um processo de automação, e documentá-lo de acordo com as boas práticas.

O objetivo é capacitar uma organização para melhor suportar a sua infraestrutura com um plano de ação proativo ou reativo em caso de alertas de falhas que interrompam os serviços IT.

A arquitetura de cada ambiente varia, mas essencialmente é necessário dispor de uma ferramenta fiável para monitorizar - um sistema de gestão central que reúne todas as informações pertinentes dos vários serviços (AD, Dynamic Host Configuration Protocol (DHCP), Internet Information Services (ISS),Windows Discovery Services (WDS), *FileShare*) em uso numa empresa. Tal pode ser feito a partir de logs, medidores de desempenho, disponibilidade dos serviços, etc. - que permitirão, a partir da informação recolhida, responder adequadamente aos eventos, conseguindo melhorar a qualidade e eficiência dos serviços prestados com o mínimo de interação humana.

Aquando da falha de um serviço ou servidor (computador) que, por algum motivo, não seja imediatamente detectado, deverá ser possível disponibilizar aos utilizadores uma interface ou ferramenta para reportar esses problemas à equipa de IT da empresa. Agilizar os mecanismos de protecção e recuperação depende do rápido diagnóstico do problema, mas também da forma uniforme e organizada de como os pedidos de suporte/incidente chegam às equipas responsáveis.

Por último podem ser acionados mecanismo automáticos de correcção/minimização dos danos utilizando uma ferramenta que coordena várias outras ferramentas. Ou seja, criar automatismos para uma separação fina dos alertas, com a criação de pedidos de incidente para as equipas com a correspondente área de ação. Naturalmente, para cada caso existe um conjunto de procedimentos e dependências que devem ser respeitadas na recuperação do serviço e no escalamento do problema à equipa responsável. Esta sequência de tarefas, se feita manualmente, é propícia a erros, pelo que se pretende implementar automatismos na distribuição dos pedidos provenientes de alertas ou de utilizadores para recuperação ou prevenção da falha de um serviço, tão completa quanto possível, e que seja simples de compreender e utilizar, permitindo também que o *feedback* das ações executadas chegue ao responsável IT.

Em suma, esta dissertação tem como objetivo dar a conhecer, e aplicar em alguns casos de estudo, um conjunto de ferramentas que estão na base da correta gestão da infraestrutura IT. Imaginando que se pretende estabelecer uma organização e respectiva área IT para suportar o negócio: este trabalho introduz alguns dos temas para os menos familiarizados com esta área, assim como pontos de possíveis melhorias na infraestrutura para a agilização das tarefas. Estas ferramentas, que são poderosas, carecem de um estudo aprofundado para evitar utilizações/configurações incorrectas que se podem tornar desastrosas para a organização (e para a área de IT).

1.3 Solução Proposta

Para muitos eventos que ocorrem na infraestrutura será possível, por análise dos mesmos, implementar mecanismos de diagnóstico e remediação de forma automatizada em caso de alerta. É expectável que seja necessário usar múltiplas tecnologias (incluindo soluções desenvolvidas para a *Cloud*) em complemento às outras vocacionadas para ambientes *on-premises* como forma a aumentar o leque de vantagens e dar visibilidade ao utilizador final das ações executadas. Ao identificar a área mais crítica ou de maior peso num ambiente heterogéneo deverá ser possível: otimizar a utilização, escalabilidade e disponibilidade dos recursos; acelerar processos de automatização para pré-requisitos e patches; identificar processos IT a melhorar; recuperar o estado de saúde dos servidores identificando serviços falhados e corrigi-los; e, como resultado, minimizar ou eliminar as operações humanas propícias à erros. O foco desta dissertação incidirá sobre tecnologias multidisciplinares da suite *Microsoft System Center*, como o Microsoft System Center Configuration Manager (SCCM), Microsoft System Center Operations Manager (SCOM), Microsoft System Center Service Manager (SCSM) e Microsoft System Center Orchestrator (SCORCH), assim como em serviços da *Azure Cloud* que são equivalentes ou se interligam às ferramentas *on-premises*.

1.4 Organização do Documento

A presente dissertação é estruturada em 7 capítulos, o primeiro dos quais é esta introdução;

O segundo capítulo discute os diferentes tipos de infraestruturas existentes, a sua evolução, diferenças e quotas de mercado. É uma importante inicialização aos diferentes tipos de ambientes com que nos podemos deparar nos clientes.

O terceiro capítulo aborda o funcionamento detalhado de uma ferramenta de monitorização principal (SCOM) em relação aos outros tipos de monitorizações, em nuvem, local ou num modelo misto.

O quarto capítulo explica a gestão de serviços segundo uma visão/abordagem Information Technology Infrastructure Library (ITIL), apresentando as vantagens e pontos de melhoria na ferramenta (SCSM), assim como integração com outras ferramentas da mesma família (SCORCH/SCOM). Neste capítulo é ainda explicado uma ferramenta de gestão central (SCCM) da infraestrutura. Apesar desta última ferramenta conter automatismos que facilitem a execução de tarefas pelos administradores de sistemas, as tarefas requerem em grande parte uma intervenção manual com salvas excepções, por exemplo, processo de atualizações e criação do processo de *deploy* de imagens de sistemas operativos.

O quinto capítulo apresenta uma ferramenta de automação/orquestração (SCORCH) que interliga-se com várias plataformas. São enumeradas algumas ferramentas do mesmo género. É também descrito o processo de automação *Cloud* ou num ambiente híbrido.

O sexto capítulo apresenta um caso real em que foram aplicadas algumas das ferramentas em estudo e a forma como estes componentes se interrelacionam de acordo com os requerimentos de clientes para uma correto funcionamento da sua infraestrutura. Em complemento foi foi um estudo de um plano de consumo de energia para máquinas postos de trabalho.

O sétimo capítulo apresenta as conclusões retiradas e, por fim, respectivas melhorias a introduzir no futuro.

Capítulo 2

A Evolução do Panorama *IT* Empresarial

As tendências empresariais estão sempre orientadas para a maximização do lucro, sendo boa parte dos gastos relacionados com a gestão/manutenção da infraestrutura IT, razão pela qual as organizações se focam bastante na redução dos custos de IT. Nesta área, os principais desafios são, figura 2.1:

Para cada uma destas áreas irão ser analisadas ao longo deste documento as diferentes soluções que melhor se adaptam e satisfazem o objetivo de diminuir a complexidade da tarefa de gerir uma infraestrutura IT.

Achamos importante começar por mostrar como as infraestruturas IT evoluíram ao longo do tempo, pois a situação atual resulta de uma composição de 3 tipos (ou fases) de infraestruturas, que irão ser apresentadas nas secções 2.1, 2.2 e 2.3.

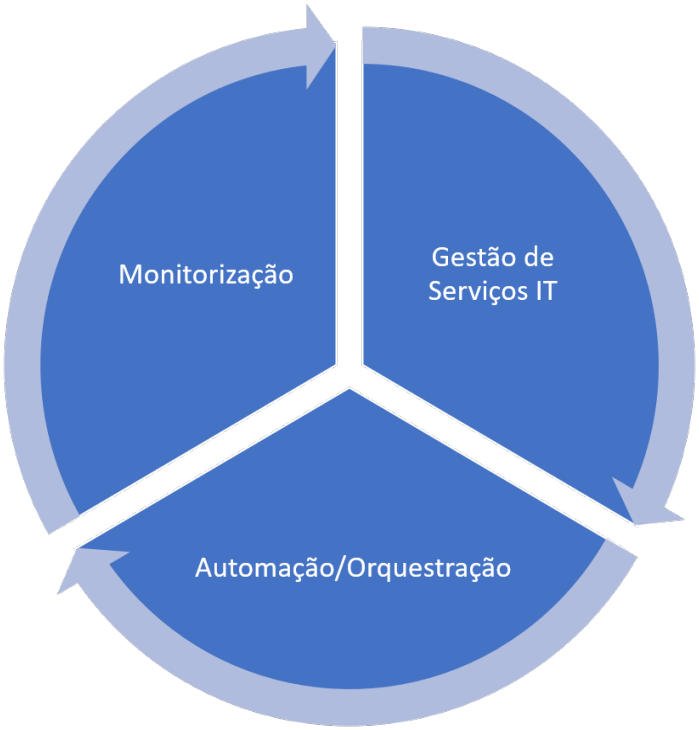


Figura 2.1: Áreas de gestão IT.

2.1 *Datacenter* sem Virtualização

Numa fase inicial, as organizações adotaram infraestruturas de IT locais, ou *on-premises* com máquinas físicas, nas quais cada servidor era destinado a um fim específico. O crescimento da infraestrutura resultante do crescimento do negócio tornou-se insustentável pela falta de espaço físico, e capacidade de refrigeração e poder de alimentação, e na qual era comum encontrar máquinas sub-aproveitadas num ou mais recursos: Central Process Unit (CPU), memória ou armazenamento (discos).

2.2 *Datacenter* com Virtualização

Numa 2ª fase, com o desenvolvimento de *hardware* mais acessível em termos de custo/eficiência (famílias x86 *Intel/AMD*) surgiu (pela mão da International Business Machines Corporation (IBM)) *software* de virtualização razoavelmente eficiente para essas arquiteturas.
A virtualização, nos termos em que nos interessa discutir aqui, é uma tecnologia que permite implementar uma abstração – máquina virtual (Virtual Machine (VM)) – de um servidor físico, possibilitando a execução de múltiplos servidores (computadores) virtuais residindo num mesmo *hardware* físico. Um hipervisor, ou Virtual Machine Monitor (VMM), é uma camada de *software* que controla os acessos das VMs ao *hardware* físico do *host*, que por sua vez suporta/hospeda a infraestrutura virtual. Existem dois tipos de

virtualização:

Tipo I - O hypervisor corre diretamente no *hardware* do servidor permitindo melhor desempenho, segurança e controlo (ver Figura 2.2). Por exemplo: *VMware ESXi*, *KVM*, *IBM z/VM*, *Citrix XenSever*, e *Microsoft Hyper-V*.

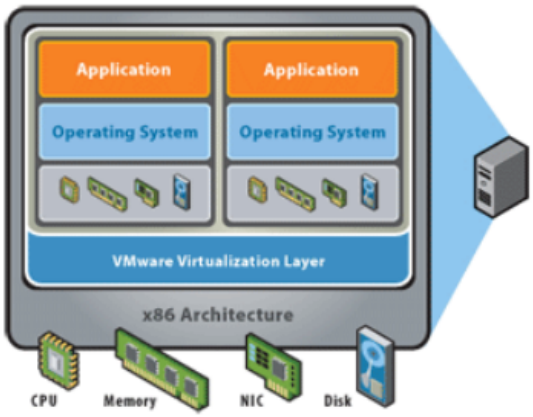


Figura 2.2: *Native/Bare-metal* hypervisor. [2]

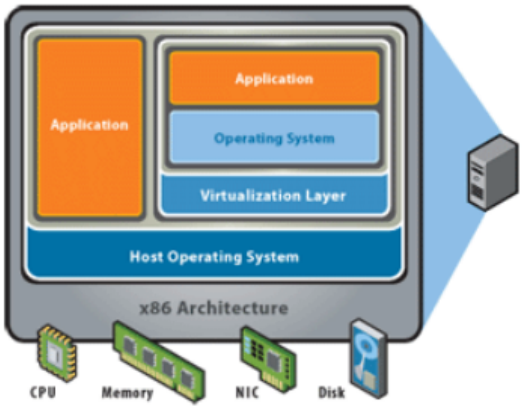


Figura 2.3: *Software/Hosted* hypervisor. [2]

Tipo II - O hypervisor corre em cima do Operating System (OS) fornecendo serviços de virtualização mas com um *overhead* maior e pior desempenho. Recomendado para um utilizador final numa máquina pessoal (ver Figura 2.3). Por exemplo: *VMware Workstation*, *Fusion* e *Oracle Virtual Box*.

A virtualização de servidores permite reduzir o espaço físico necessário para suportar a infraestrutura assim como introduzir automatismos de gestão, facilitando teste de aplicações e a migração de serviços ou até para recuperação em caso de desastre. Permite ainda a rápida criação de novas VMs por clonagem, a preservação do estado instantâneo de uma VM por meio de *snapshots*, a migração de VMs entre servidores, etc..

2.3 *Cloud Computing*

Mais recentemente (3ª fase), com o surgimento dos serviços *Cloud Computing* observou-se a adopção acentuada pelas organizações destes mesmos serviços. As empresas que já tinham um grande investimento feito numa infraestrutura local transformaram-na num infraestrutura híbrida, e para as novas empresas, estas puderam convenientemente optar por investir numa infraestrutura totalmente na nuvem.

De acordo com National Institute of Standards and Technology (NIST), "*Cloud Computing* é um modelo que permite o acesso a pedido, ubíquo, conveniente, suportado na rede, a um grupo de recursos computacionais configuráveis (e.g. redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente aprovisionados e libertados com o mínimo de esforço, gestão ou interação" [3] entre o consumidor e o provedor do serviço.

Este modelo *cloud* caracteriza-se por cinco propriedades base (ver tabela 2.1), três modelos de implantação (ver secção 2.3.1) e três modelos de serviço (ver secção 2.3.2).

Tabela 2.1: As 5 Propriedades da *Cloud*.

On-Demand Self-Service	Um consumidor pode aprovisionar unilateralmente e automaticamente com os recursos de computação que acha necessário, sem a necessidade de interação humana com o provedor de serviços.
Broad Network Access	Os recursos estão disponíveis na rede e são acedidos via protocolos standard disponíveis nas diferentes plataformas cliente.
Resource Pooling	Os recursos do provedor cloud são agrupados para servir múltiplos consumidores (Multi-tenant model), estes recursos físicos/virtuais e podem ser dinamicamente alocados e realocados de acordo com o nível de procura dos serviços.
Rapid Elasticity	Os recursos podem ser dinâmica/elasticamente (de-) aprovisionados, preparados para automaticamente e rapidamente escalar quando necessário, de forma transparente para o consumidor.
Measured Service	Os sistemas *Cloud* devem automaticamente medir, controlar, otimizar e reportar os recursos/serviços utilizados tanto para o consumidor como provedor.

2.3.1 Os diferentes modelos de implantação de uma *Cloud*

Do ponto de vista da implementação (Deployment), uma infraestrutura em nuvem pode ser dividida em três categorias:

⇒ Na **Cloud privada** os recursos tecnológicos (computação, armazenamento e rede) são utilizados exclusivamente por uma empresa ou organização; a sua gestão é em geral efectuada diretamente pelo departamento de informática da organização, podendo a infraestrutura ser alojada interna ou externamente num provedor de serviços de *hosting*, mas com o *hardware* dedicado. No entanto, os serviços e a infraestrutura são sempre mantidos numa rede privada, e o *hardware* e *software* são dedicados exclusivamente à organização. Desta forma, esta pode usufruir das vantagens de personalizar o ambiente para corresponder às necessidades empresariais específicas e, ao não partilhar os recursos, aumenta os níveis de segurança, controlo e boa escalabilidade quando bem planeada. [2, 4]

⇒ Na **Cloud pública** os recursos são detidos e operados por um fornecedor de serviços *Cloud* externo (ver Figura 2.4) e disponibilizados através da Internet. Recursos físicos como o *hardware*, armazenamento e dispositivos de rede são partilhados com outras organizações – Multi-tenant model. Um dos benefícios típicos é o modelo de facturação, que permite pagar somente pelo serviço que se utiliza; a este acrescem

a ausência de necessidade de manutenção por parte do consumidor, ou de preocupações com as questões de escalabilidade e fiabilidade, que são "garantidas"pelo provedor. [2, 4]

⇒ Contudo, nas organizações de média/grande dimensão observa-se, na prática, a combinação das duas *Clouds* anteriores tirando partido do "melhor dos dois mundos": as **Clouds híbridas** surgem da adoção de partes da *Cloud* pública pelas organizações que já tem um investimento numa infraestrutura física local. Os dados e as aplicações podem mover-se entre *Clouds* privadas e públicas para uma maior flexibilidade e personalização. Por exemplo, utilizar a *Cloud* pública para necessidades com grande volume e baixa segurança, e a *Cloud* privada para operações confidenciais e críticas para a negócio, ou simplesmente, estender a infraestrutura para a *Cloud* pública aquando aos picos de utilização. [2, 4, 5]

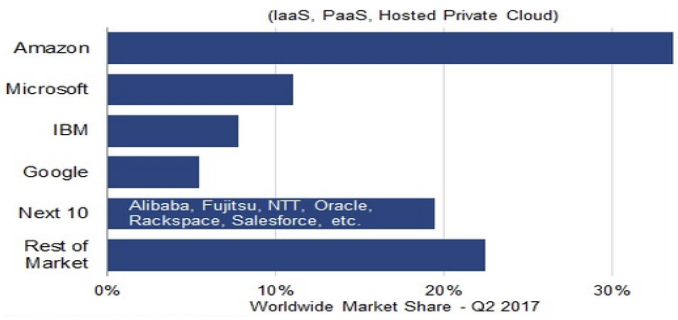


Figura 2.4: Principais fornecedores de serviços *Cloud*. [6]

2.3.2 Os diferentes modelos de serviço de uma *Cloud*

Os três principais modelos de serviço *Cloud* (ver Figura 2.5) definem a que nível se posiciona o consumidor do serviço, a flexibilidade que pode ter na parametrização do mesmo, e o esforço que tem de despender na sua gestão.

⇒ **Infrastructure as a Service (IaaS)** cabe ao consumidor fazer o (a)provisionamento das VMs, definindo a sua capacidade de processamento (número de virtual Central Process Units (vCPUs)), de armazenamento (quantos discos e com que capacidade), de virtual Random Access Memory (vRAM), de interfaces de rede (virtual Network Interface Cards (vNICs)), qual o sistema de operação *guest*, e *software* a instalar [3]. O Amazon Web Services (AWS) Elastic Compute Cloud (AWS) (EC2) e o *Azure Virtual Machines* são dois exemplos de serviços de nível IaaS de *clouds* públicas. [2, 4]

⇒ **Platform as a Service (PaaS)** cabe ao consumidor implantar na infraestrutura *cloud* as diferentes aplicações que desenvolveu, recorrendo aos serviços já disponibilizados pela plataforma, e evitando assim o *deployment* de serviços já existentes. Por

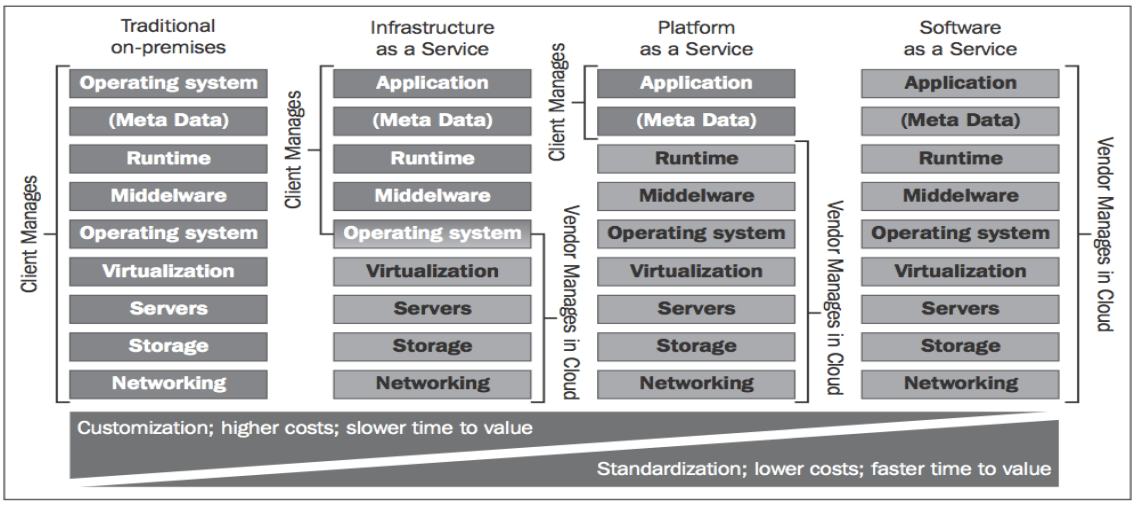


Figura 2.5: *On-Premises* & Modelos de serviços *Cloud*. [7]

exemplo, usando o serviço *MySQL* já disponibilizado pelas plataformas *Azure Databases* ou *AWS DynamoDB*. Estes serviços PaaS têm ainda a vantagem de serem totalmente geridos em termos de atualizações e até *backups* pelos provedores, pelo que o consumidor apenas tem de se preocupar com "adquirir"o nível de desempenho/qualidade de serviço desejado. [2, 4]

⇒ **Software as a Service (SaaS)** é o modelo mais indicado para as organizações reduzirem custos e reagirem às oscilações na procura do seu serviço. Os utilizadores acedem e usam aos serviços diretamente do *browser*, não necessitam de instalar aplicações, ou preocuparem-se com o OS, servidores, armazenamento de dados, *backups*, etc. Por exemplo: *Microsoft Office 365*, *Google Gmail*. [2, 4]

2.4 Mover para a *Cloud* ou ficar *On-Premises*?

Chegados a este ponto, abandonamos a ideia de que *Cloud Computing* é apenas virtualização, sendo antes uma forma de utilizar a tecnologia como um serviço. Os consumidores, em particular os que usam PaaS ou SaaS, precisam de ter pouco ou até nenhum conhecimento dos detalhes de como um serviço em particular é implementado, como por exemplo em que *hardware* ou qual o número de CPUs em que é executado, dando oportunidade de reduzir a complexidade do que é montar e gerir uma infraestrutura. O único fator importante é ter um bom entendimento de como o serviço funciona e usá-lo adequadamente no portal. Deste modo, o consumidor paga o que usa, beneficiando dos princípios fundamentais sobre os quais a *Cloud Computing* está construída:

1. Segurança;
2. Desempenho e Escalabilidade;
3. Disponibilidade e Recuperabilidade;

4. Eficiência de Funcionamento.

Por razões de segurança, privacidade e integridade muitas entidades públicas/financeiras ou empresas de maior dimensão não têm a liberdade total para usufruir das vantagens dos modelos *Cloud*, a fim de não arriscarem comprometer dados sensíveis de negócio.

Recentemente, estas restrições sobre o paradigma *Cloud* tem vindo a mudar com a entrada do *Azure Stack* da *Microsoft*. O *Azure Stack* é uma oferta que inclui *hardware/-software* especializado que é instalado no *datacenter* de uma organização, e permite ter os mesmos serviços *Azure* da *Cloud* pública alojados a correr na própria infraestrutura do cliente. O *Azure Stack* é uma ponte entre a *Cloud* privada e pública e permite a escalabilidade dos recursos quando é necessário estender a infraestrutura para a *Cloud* pública. O que diferencia *Azure Stack* dos competidores com produtos semelhantes, por exemplo *AWS Outpost*, é a possibilidade de correr totalmente isolado e desconetado da *Cloud* pública, protegendo a privacidade e integridade dos dados. Contudo, ainda é necessário um amadurecimento desta tecnologia para suportar todos os serviços *Cloud*, como Disaster Recovery (DR), e fazê-lo a um preço mais acessível.

Capítulo 3

Monitorização Centralizada

A monitorização dos sistemas e serviços da infraestrutura é uma tarefa crítica para o administrador de sistemas. Deverá ser possível ter uma visão em tempo real do estado de utilização e disponibilidade dos recursos, por exemplo, tráfego da rede, espaço em disco, número de núcleos (CPU e/ou cores) e memória Random Access Memory (RAM) utilizada. Dependendo do tipo de infraestrutura (*on-premises*, *cloud* ou híbrida) há soluções que melhor se adaptam a cada uma, e que irão ser analisadas, relativamente à sua arquitetura e vantagens, ao longo deste capítulo.

3.1 *On-premises*

As ferramentas de monitorização começaram por ser desenvolvidas para sistemas tradicionais *on-premises*, e têm por isso mais tempo de maturidade e aperfeiçoamento; destacamos as seguintes:

- *DataDog*;
- ***Nagios***;
- ***Microsoft System Center Operations Manager (SCOM)***;
- *SolarWinds*;
- *Zabbix*;
- *Zenoss*;

3.1.1 *Nagios*

Nagios é um *software*, *open-source* na sua versão base, muito popular na monitorização de sistemas *Unix/Linux* e *Windows*, que permite fazer verificações de aplicações, redes e serviços. Além de poder ser operado a partir de linha de comandos, tem também uma

interface gráfica *Web* com visualização em *dashboards* dos vários alertas por níveis de gravidade, baseados em parâmetros estabelecidos, onde é possível emitir automaticamente alertas por *email* ou mensagens texto [8].

Existe um *Management Server (Unix/Linux)* que recebe os dados reportados pelos agentes instalados nas máquinas. Contudo, também permite uma monitorização *agentless* (emulação de um agente) para recolher métricas de *switches*, *routers* ou serviços acessíveis por protocolos de rede. É configurado a partir de um ficheiro principal onde é possível mapear e definir as configurações. [9]

Apesar de uma reputação com mais de 20 anos, a usabilidade e sofisticação do produto continua muito rudimentar e necessita de um nível de especialização elevado. Num ambiente moderno, o *Nagios* revela-se inconsistente na monitorização de grandes infraestruturas ou infraestruturas com grande dinamismo – constante entrada e saída de máquinas. O *upgrade* ou a manutenção das configurações é trabalhoso e torna-se difícil alcançar os objetivos quando o conhecimento está restrito a um grupo específico de pessoas.

3.1.2 *Microsoft System Center Operations Manager*

SCOM ou *OpsMgr* é um *software* que faz parte da suite *System Center* da *Microsoft*. *System Center* engloba vários produtos de monitorização e gestão de aplicações, ambientes físicos ou virtuais, cada um com soluções para objetivos específicos. Apesar de cada produto ter uma licença comercial e poder ser adquirido individualmente, se interligado com outros produtos da mesma suite proporciona uma gestão uniformizada acrescida.

Similarmente ao *Nagios*, SCOM é uma ferramenta de monitorização de servidores, postos de trabalho, aplicações e ativos de rede, entre outros equipamentos que, quando agregados em grupos permite uma visão simplificada em tempo real do ambiente monitorizado. Agrega numa única interface, baseada em agentes instalados em máquinas *Windows* e *Unix/Linux*, a informação e alertas definidos de acordo com a disponibilidade, performance, configuração e segurança. Integra comandos *PowerShell* para facilmente diagnosticar problemas. Estas capacidades permitem às equipas de administração e operação de sistemas reagirem pro-ativamente na identificação e resolução de eventos que comprometam ou possam vir a comprometer os serviços disponibilizados pelo IT.

Imaginando um cenário no qual um dos discos de um servidor que guardam os *logs* numa base de dados SQL está na capacidade máxima, apesar do servidor "não ficar danificado"deixa de ser possível efetuar novas operações, interrompendo por completo o serviço. Neste cenário podemos categorizar o estado do disco em quatro categorias:

- P0 - Cor vermelha (alerta crítico), indica que o espaço ocupado em disco é superior a 90%;
- P1 - Cor amarela (alerta médio), indica que o espaço ocupado está entre 70% a 90%;
- P3 - Cor verde, indica que o espaço ocupado é inferior a 70%;

- P4 - Cor cinzenta, não é possível obter o estado do disco (sem comunicação ou não disponível).

Dado uma classe de alertas, é possível criar canais de notificação sobre o tipo de alerta e delegar responsabilidades às equipas IT, e automaticamente definir uma tarefa de manutenção à uma máquina defeituosa. Deste modo, a equipa notificada, ao receber um alerta P1, rapidamente poderá adicionar mais espaço em disco ou limpar o ficheiro de *log* evitando a interrupção do serviço.

Além da vista de alertas, existem outras vistas importantes como a de desempenho e *dashboards*, onde se sumarizam e se agregam as informações em tempo real do estado da infraestrutura, recolhidas à partir de monitores e regras. Estes contadores agem sobre um grupo de instâncias, de uma ou mais classes-tipo, por exemplo, um grupo que contém os membros da classe *Windows Server 2019 - Logical Disk* e da classe *Windows Server 2019 - Network Interface* contém todos os discos rígidos e placas de rede que forem adicionados às máquinas da infraestrutura.

O anexo I lista um exemplo de um *PowerShell script* que desenha o *PowerShell Grid Widget* da figura 3.1 para mostra o desempenho do disco.

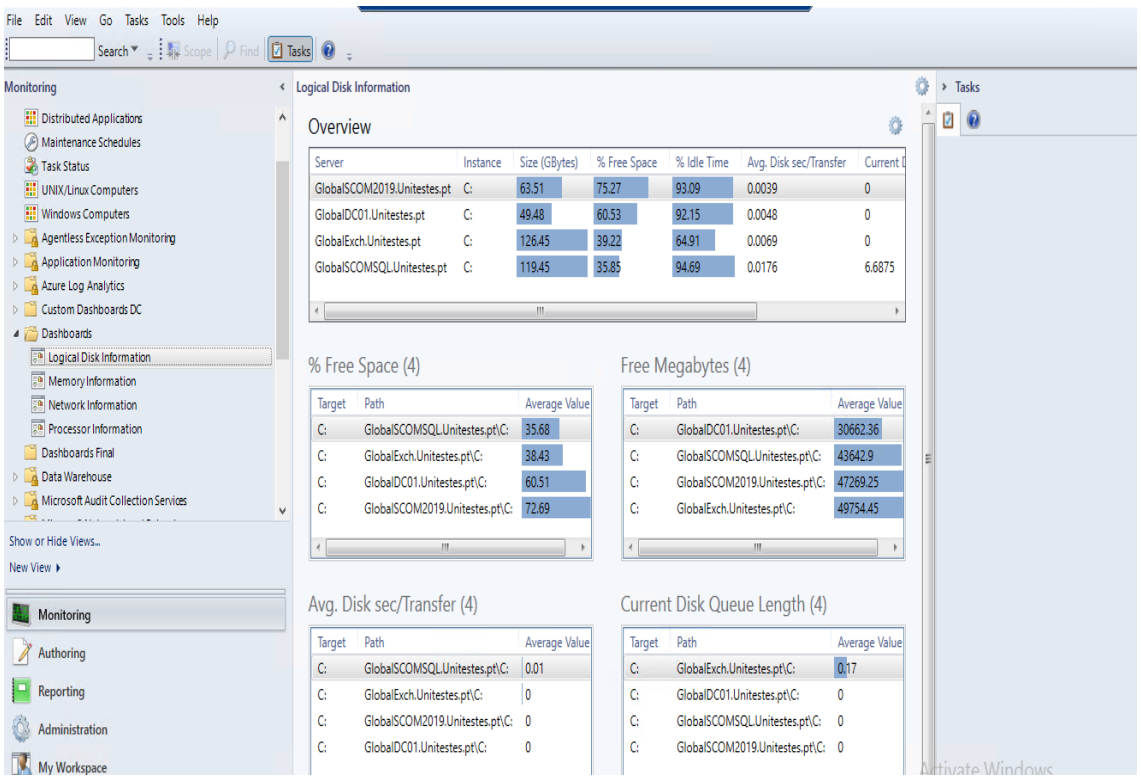


Figura 3.1: Exemplo *PowerShell Dashboard script*.

A arquitetura do SCOM, exibida na figura 3.2, é composta pelo [10]:

- ***Management Server*** (servidor central) com a função de monitorização, tratamento de regras e de monitores, assim como das consolas de gestão e de aplicações distribuídas. É recomendado, dependendo da dimensão da infraestrutura, mais do que

um *Management Server* para distribuição de carga (de vários agentes) e continuidade do serviço em caso de falha, tornando a infraestrutura de monitorização redundante. Alberga uma base de dados responsável pela informação dos objetos monitorizados. É importante salientar que apesar de teoricamente ser possível ter um servidor central *on-premises* e outro na nuvem, esta abordagem apresenta problemas de desempenho e foge às boas práticas, pois a comunicação entre os dois servidores de gestão requer uma latência inferior a 5ms.

- ***Operational Database*** é a base de dados do SCOM que contém a informação de configuração, segurança e operações dos objetos da infraestrutura;
- ***DataWarehouse Database*** é a base dados do SCOM que contém o histórico (informação à longo prazo) dos alertas, configurações, segurança, dados de desempenho recolhidos e operações realizadas sobre os agentes;
- ***Consola*** é a interface gráfica onde estão apresentados os eventos relativos aos objetos monitorizados. As vistas da consola podem ser personalizadas de acordo com o perfil de utilização ou às diferentes equipas de gestão.
- ***Web Console*** alojada num *Management Server* pode ser acedida através do *browser*, caso não exista uma consola instalada, facilita o acesso e consulta do conteúdo alarmístico;
- ***Reporting Server*** tem como função recolha e processamento de informação para gerar relatórios da atividade de monitorização com base nos dados da *DataWarehouse*. Os relatórios podem ser customizados e distribuídos por *email*, informando dos dados de desempenho, disponibilidade e alarmística;
- ***Gateway Server*** é um servidor intermediário entre o *Management Server* e os agentes de um outro domínio. A comunicação segura e autenticada é estabelecida entre a *Gateway* e o *Management Server* através de certificados e encriptação;
- ***Audit Collection Service*** recolhe eventos das máquinas para efeitos de *compliance*, por exemplo falhas de *login* e outros eventos achados pertinentes;
- ***Agent*** é a unidade instalada numa máquina *Windows* ou *Unix/Linux*, que recolhe os métricas de desempenho, gerando eventos baseado neles;
- ***Agentless*** adequado aos objetos a monitorizar em que não seja possível instalar o agente, é necessário um servidor que faça de *proxy* para direcionar a informação recolhida.

Para um correto funcionamento de uma infraestrutura de SCOM é necessário assegurar a conetividade de rede num conjunto de portas de comunicação Transmission Control Protocol (TCP) e User Datagram Protocol (UDP). Todas as comunicações iniciadas pelos agentes e pelos *Management Servers* são efetuadas através do Fully Qualified Domain Name (FQDN) do equipamento alvo, ou seja, todos os servidores/agentes terão de conseguir resolver o IP/FQDN de cada um.

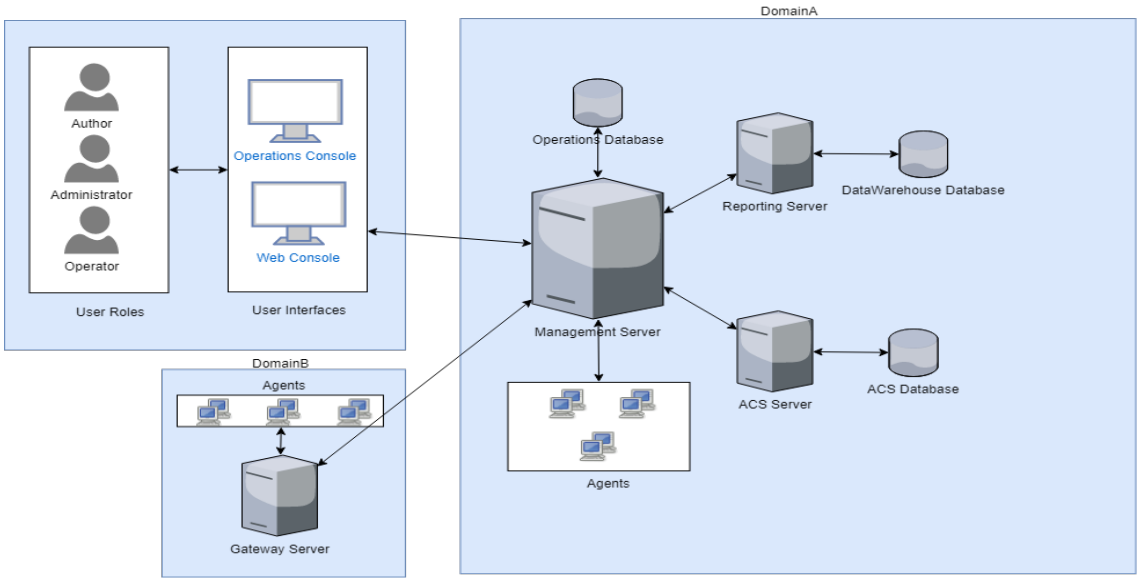


Figura 3.2: Arquitetura *SCOM*.

Para uma infraestrutura de média/grande dimensão, o estudo dos pré-requisitos, abertura de conetividades (porta 5723 para comunicação, porta 5724 para a consola SCOM, entre outras), a criação de contas e a instalação é um processo bastante complexo com uma duração estimada de cerca de uma semana (dependendo do tamanho da infraestrutura). Uma vantagem na instalação dos agentes é ser feita diretamente à partir do *Management Server* sem ser preciso conectar-se à cada máquina agente.

3.1.2.1 Monitorização

Sendo a principal componente do SCOM, é uma vista que trata de monitorizar constantemente os vários agentes da infraestrutura tecnológica por via de "health explorer"ou gráficos de desempenho das mesmas.
Esta monitorização é realizada através de diversos monitores e regras que são utilizadas para analisar informação relevante e transmitir esses dados ao servidor de SCOM, comunicando na porta 5723. Por sua vez, estes monitores e regras, são encapsulados no que se chama de Management Pack (MP). Um MP é um agregador de regras, monitores e outros objetos, normalmente relacionados entre si.

O módulo de monitorização, para além da identificação de eventuais problemas, contém ainda informações e tarefas de diagnóstico e resolução dos mesmos através de processos que podem ser automatizados.

Resumindo, é neste módulo (ver figura 3.3) que se fazem as seguintes tarefas:

- Gestão de alertas;
- Monitorização de objetos (máquinas agentes);
- Execução de tarefas de diagnóstico e recuperação;

- Possibilidade de obtenção de informação relativa a um evento ou alerta específico, com acesso direto à base de dados de conhecimento da *Microsoft*;
- Recurso à função de manutenção para clientes alvo de intervenção programada;
- Visualização de diagramas da infraestrutura.

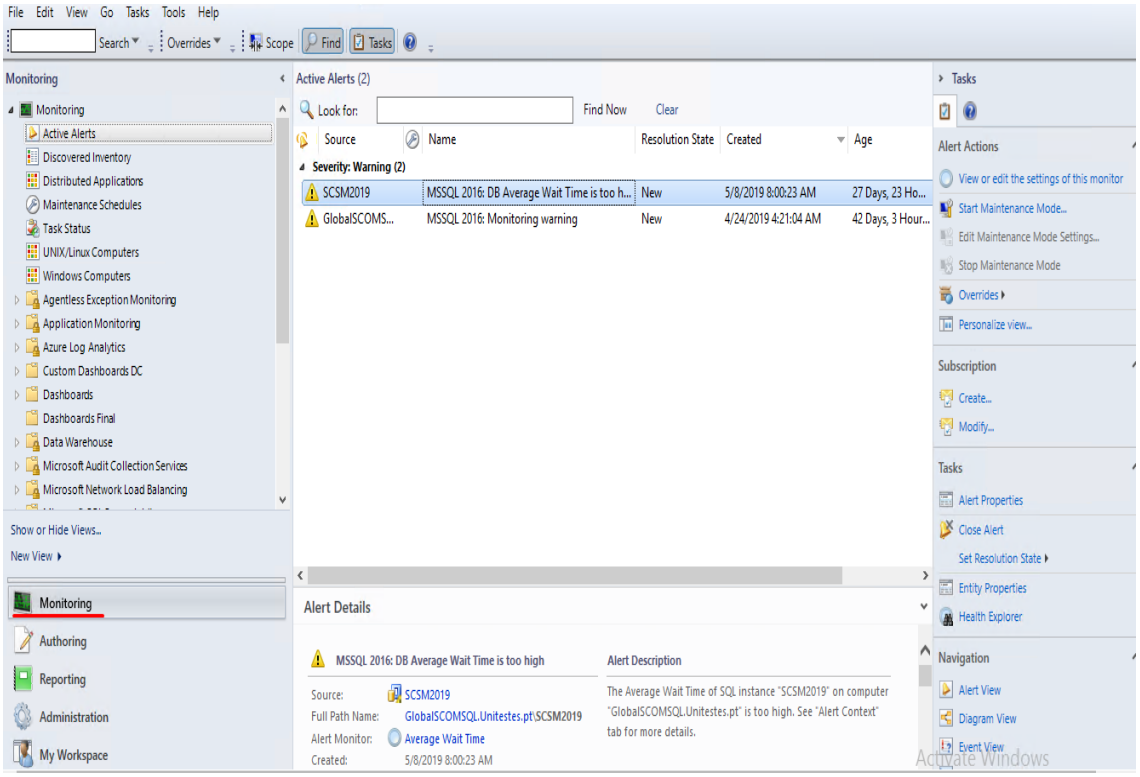


Figura 3.3: Vista de Monitorização.

3.1.2.2 Authoring

Outro dos módulos disponíveis na consola de SCOM é o módulo de *Authoring* (ver figura 3.4). Este módulo divide-se em duas áreas, que nos permite criar ou alterar objetos de monitorização dos MP, como por exemplo, monitores e regras. Podemos criar aplicações distribuídas, grupos e monitorizações específicas na disponibilidade de serviços, conexões de SQL, *Web Applications*, etc. A criação de grupos dinâmicos baseados nas propriedades dos objetos com a(s) mesma(s) propriedade(s) é uma requisito para a correta monitorização e afinação dos alertas gerados.

Qualquer atividade de *Authoring* só pode ser diretamente efetuada sobre objetos pertencentes a MPs não selados. Quando o MP se encontra selado, este não é diretamente editável tendo de ser criado um novo MP para alojar os dados ou alterações pretendidas.

O *Override* é utilizado para alterar o comportamento de monitores, atributos, regras, descoberta de objetos e *thresholds* e pode ser aplicado a um objeto, a um grupo de objetos, a uma classe ou um objeto dessa classe. Com o recurso a *overrides*, torna-se possível ajustar à realidade de cada organização a forma como esta é monitorizada.

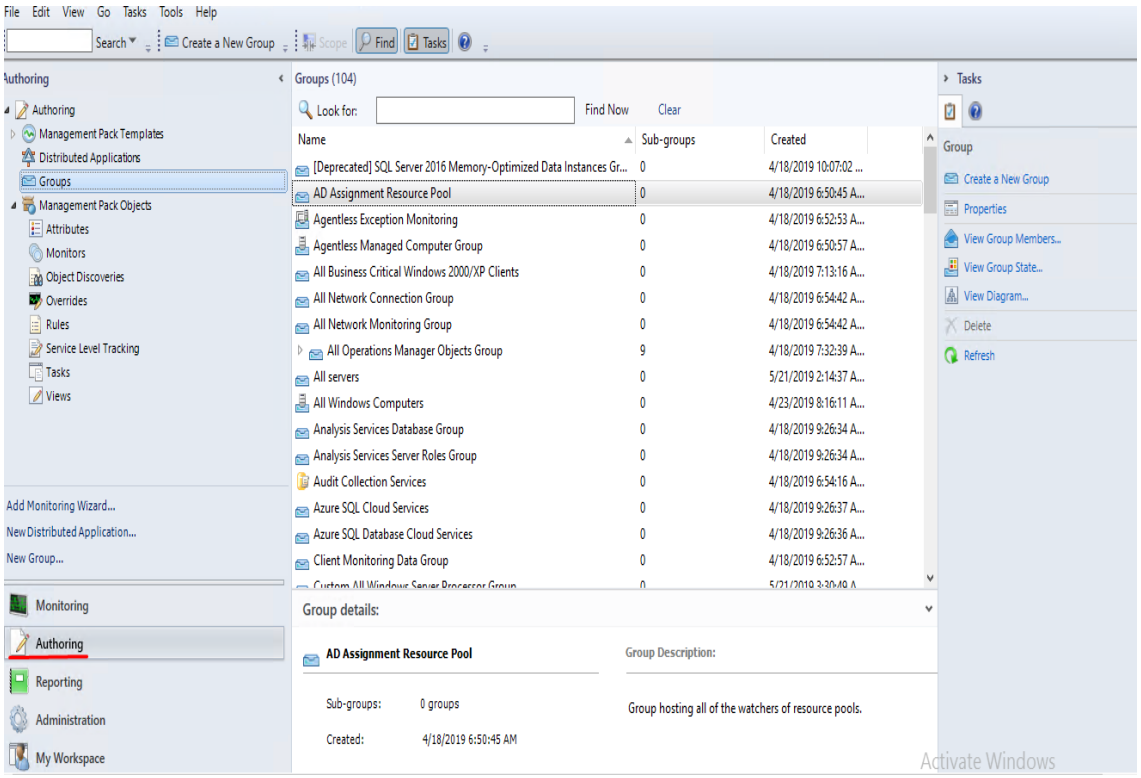


Figura 3.4: Vista de Criação.

3.1.2.3 Reporting

O módulo de *Reporting* permite o acesso ao histórico de dados operacionais. Cada MP introduz neste módulos relatórios pré-configurados, que permitem pesquisar informação baseada em diversos critérios como a data, objetos ou grupos relevantes, sendo possível neste módulo:

- Configurar relatórios com base na informação dos diferentes MP;
- Agendar a criação de relatórios;
- Exportar relatórios;
- Calendarizar envio de relatórios por email.

3.1.2.4 Administration

O módulo de Administração do SCOM (ver figura 3.5) servirá para que os administradores de sistemas possam efetuar a configuração e manutenção do produto, sendo possível realizar as seguintes tarefas:

- Gerir e configurar o SCOM;
- Importar, exportar e atualizar MP;
- Instalar, reparar, atualizar e remover agentes de SCOM pelos diferentes *Management Servers* ou *Gateways*;

- Gerir privilégios de segurança;
- Gerir e configurar descoberta de ativos de rede;
- Configurar subscrições e notificações de alertas.

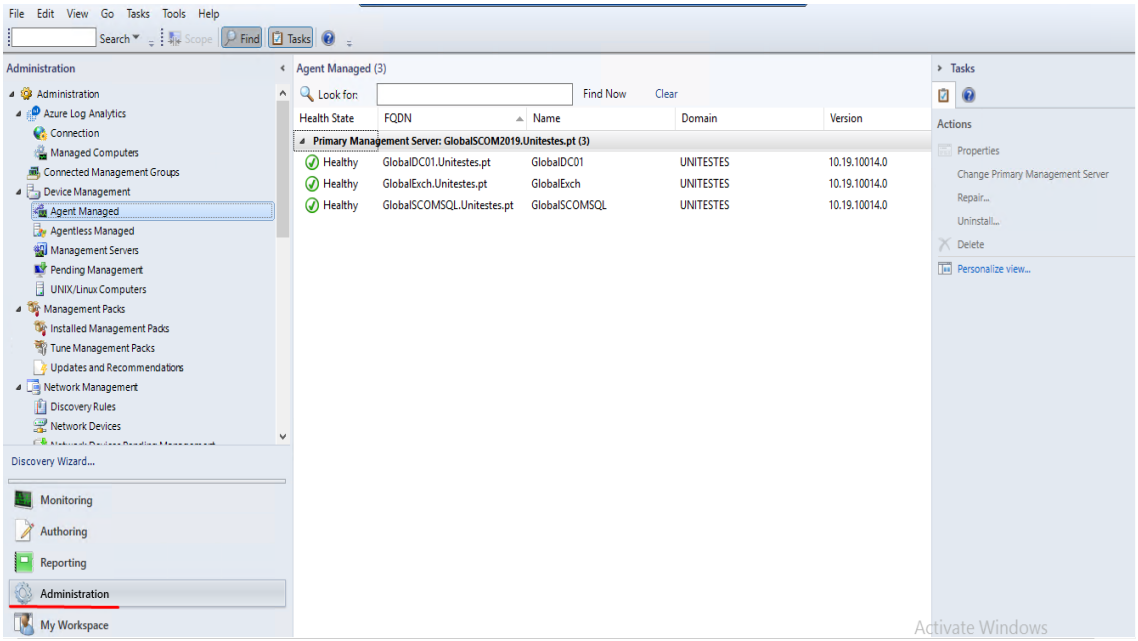


Figura 3.5: Vista de Administração.

Para efeitos de separação de funções de administração no SCOM, existem diversos perfis de utilizador previamente definidos. Estes perfis e correspondente descrição encontram-se na tabela 3.1:

Tabela 3.1: Lista de Perfis de Utilizador.

Administrador	Controlo sobre todas as permissões do SCOM.
Operador	Interação com alertas e execução de tarefas em interfaces específicas.
Operador Avançado	Permissões do perfil Operador, com possibilidade de configurar regras e monitores. Pode ser limitado a um grupo específico de monitorização.
Operador de Leitura	Acesso de visualização a uma consola de alertas. Pode ser limitado a um grupo específico de monitorização.
Operador de Relatórios	Acesso somente à funcionalidade de *Reporting*. Pode ser limitado a um grupo específico de objetos.
Operador de Monitorização Aplicacional	Acesso de visualização de alarmística sobre Diagnósticos Aplicacionais.
Autor	Criação, edição e/ou remoção de tarefas, regras, parâmetros de monitorização e interfaces. Pode ser limitado a um grupo específico de objetos.
Administrador de Segurança de Relatórios	Define o controlo de acesso ao módulo de *Reporting*.

3.1.2.5 My WorkSpace

O módulo *My Workspace* tem como utilidade guardar customizações no perfil de cada utilizador/administrador, permitindo guardar vistas específicas no ambiente de cada um, apenas apresenta os itens necessários para que cada um desenvolva o seu trabalho no SCOM, em analogia, são os favoritos de cada utilizador.

3.1.2.6 Management Packs

Uma das vantagens do SCOM é a interligação com várias plataformas e aplicações através dos MPs. Os MPs são ficheiros Extensible Markup Language (XML) que guardam as regras de monitorização pré-definidas, no entanto customizáveis para satisfazer requisitos funcionais e de negócio (*Overrides*), para aplicações ou sistemas alvo e comtém referências para outros MPs selados. As regras de monitorização sobre o estado da máquina são categorizadas quanto à disponibilidade, configuração, desempenho e segurança.

Os elementos que compõem um MP, ver figura 3.6. Nos sub-pontos seguintes descrevemos os componentes mais importantes.

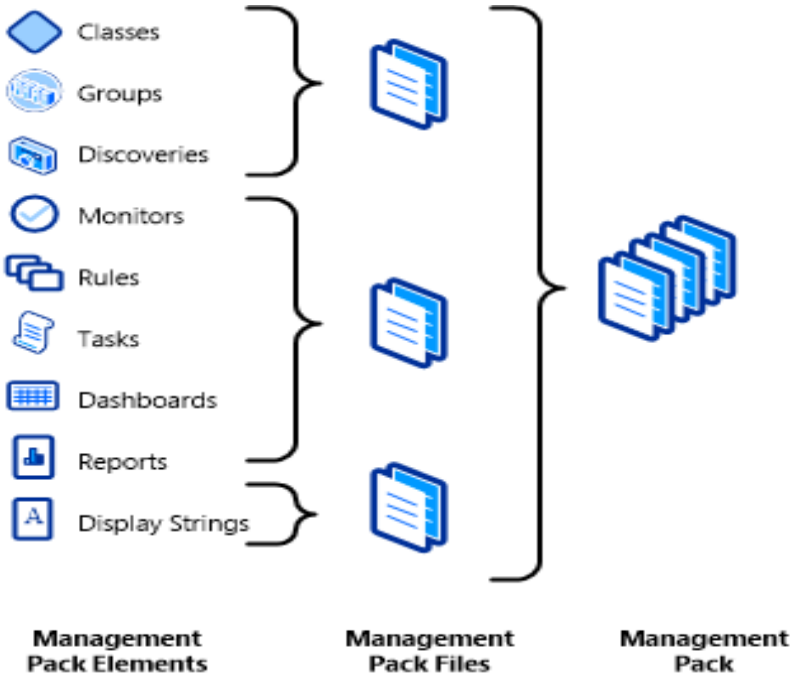


Figura 3.6: *Management Pack Model*.[11]

Classes A classe representa um tipo de objetos e cada objeto no SCOM representa uma instância de uma classe. As instâncias contêm um conjunto de atributos próprias daquela classe. Cada atributo tem o seu próprio valor determinado na descoberta do objeto. Um MP define uma série de classes com os seus atributos e relações entre essas classes. Todas as classes têm uma classe base ou classe "mãe"e seguem os mesmos conceitos de herança

e polimorfismo que a programação orientada à objetos[12]. Analisar a figura 3.7 para um entendimento do modelo das classes no SCOM. [13]

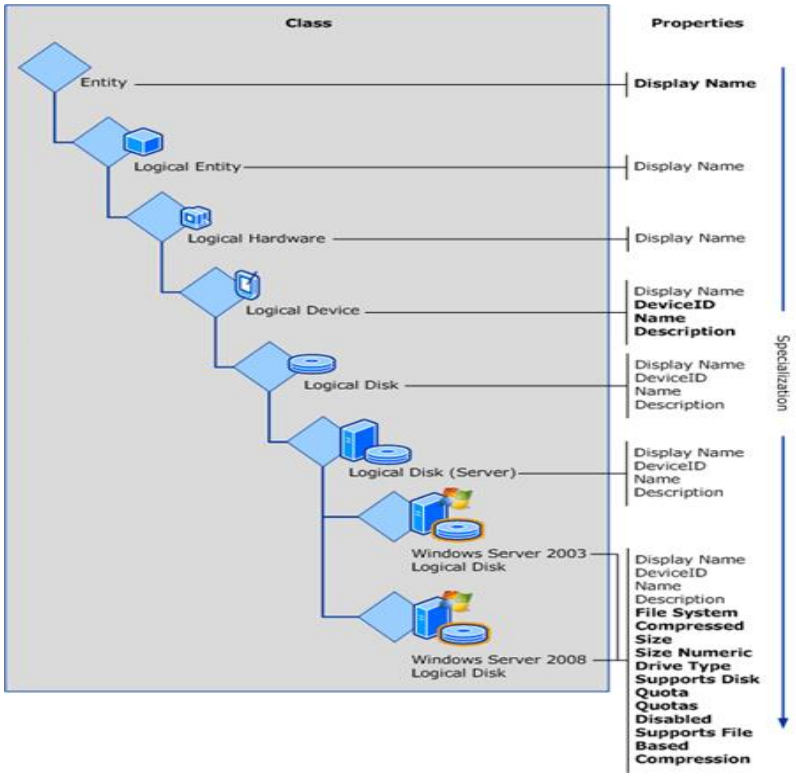


Figura 3.7: *SCOM Class Model*.[12]

Monitores Os monitores são utilizados para recolher informação dos objetos monitorizados podendo, por exemplo, medir a desempenho ou detetar a ocorrência de um evento. Do resultado desta recolha, surgirá a definição do estado do monitor - *health state*. Para cada monitor existem quatro estados possíveis: *Healthy*, *Warning*, *Critical* e *Unmonitored*. Cada um destes estados pode ser definido e alterado consoante a criticidade do objeto alvo de monitorização e dos valores dos parâmetros envolvidos. Existem três tipos de monitores:

- ***Unit Monitor***: utilizado para monitorizar contadores, eventos, serviços e scripts específicos; o seu estado pode ser propagado para os outros dois tipos de monitores e podem ser configurados para emitir um alerta na ocorrência de uma mudança de estado; [14]
- ***Aggregate Rollup Monitor***: utilizado para refletir o estado de um componente tendo como base a percentagem dos monitores pertencentes ao componente a que o *Aggregate Rollup Monitor* pertence; [15]
- ***Dependency Rollup Monitor***: este monitor reflete o estado de outros objetos que possuem uma ligação ao objeto monitorizado, seguindo a lógica de monitorização de serviço; [15]

Descoberta de objetos Neste ponto podemos definir a descoberta de objetos referentes aos equipamentos alvo de monitorização. Esta descoberta é dinâmica e pode utilizar o registo dos clientes, Windows Management Instrumentation (WMI), *Scripts*, Object Linking and Embedding (OLE) DataBase (DB), Lightweight Directory Access Protocol (LDAP), Simple Network Management Protocol (SNMP) para descobrir diferentes tipos de objetos numa rede corporativa.

Regras O SCOM utiliza as regras de monitorização para determinar como deve recolher, processar e responder à informação gerada pelos agentes baseada em *thresholds*. As regras são responsáveis pela reação imediata a um conjunto de eventos, para responder de forma ativa no caso de falha, através da execução de um *script* para o evento em questão. Estas regras permitem agir de forma inteligente e pró-ativa relativamente a um conjunto padrão de eventos, despoletando ações ou gerando alertas administrativos. [16]

As regras fazem ainda a ligação de um conjunto de eventos a uma série de artigos da *Microsoft Knowledge Base*, providenciando de imediato uma possível solução para o problema em questão.

Os diferentes tipos de regras e fontes de dados que podem ser utilizados no SCOM são os seguintes:

- Regras para gerar alertas recorrendo a uma das seguintes fontes de eventos: ficheiros de *log* genéricos (Comma-Separated Values (CSV)), log de eventos New Technology (NT), SNMP *Traps*, *Syslog*, eventos WMI;
- Regras de recolha de dados utilizando uma das seguintes fontes de contadores de desempenho: SNMP, WMI, *Windows*;
- Comandos: execução de um comando ou de um *script*.

Tarefas As tarefas ou *Tasks* permitem a execução de ações sobre o objeto monitorizado. Para cada tipo de objeto existe um conjunto de tarefas pré-definidas configurados em cada MP, e que podem ser alargados através da adição de novas tarefas personalizadas. [17]

Vistas As vistas apresentam informação sobre os objetos monitorizados, com base nos dados recolhidos em tempo real. O conteúdo de uma vista muda consoante o tipo de objeto a monitorizar e os critérios definidos na sua criação e podem ser de vários tipos, como por exemplo vistas de desempenho, estado dos objetos, lista de alertas, *dashboards*, entre outros.

Notificações O SCOM permite que sejam configuradas notificações com o objetivo de alertar de forma mais rápida os responsáveis operacionais na ocorrência de um determinado evento, podendo corrigir o problema de uma forma mais eficiente, reduzindo o tempo de uma eventual quebra de serviço.
Uma notificação pode ser configurada tendo em conta diversos critérios como a criticidade, desempenho de contadores ou até mesmo o tempo de criação de um alerta.

Contas de execução do SCOM Existem diversos tipos de contas adequadas aos tipos de tarefas. Cada um destes tipos de contas encontra-se descrito em baixo:

- **Action account**: utilizada para recolher informação dos agentes; é a conta utilizada pelo agente de monitorização, sendo responsável por executar ações que estejam pré-configuradas, pelo que deve ter privilégios de administração local em cada um dos clientes servidores;
- **Data Warehouse Writer account**: utilizada para escrever informação nas bases de dados do SCOM;
- **Data Warehouse Reader account**: utilizada para obter informação das bases de dados do SCOM a serem utilizados nos *reports*;
- **SDK Account**: utilizada para inicializar os serviços de *Windows* e interligar o SCOM e a consola à Base de Dados.

3.2 *Cloud*

Para a computação em nuvem analisou-se a *Azure Cloud* da *Microsoft* e *OpenStack* e identificou-se os serviços responsáveis pela monitorização do estado/uso dos recursos.

3.2.1 *Azure*

A *Cloud Azure* além de contar com as inúmeras vantagens sobre os serviços que oferece engloba soluções que permitem ter uma visibilidade *end-to-end* dos recursos *Azure* utilizados sob forma de gráficos, *dashboards* e alertas personalizados, nomeadamente:

- **Azure Monitor** - métricas, *logs*, eventos sobre a infraestrutura (+ de 30 serviços) como as VMs, *Web Servers*, *Storage* e *Load Balancer*, é possível definir alertas personalizados;
- **App Insight** - métricas sobre as aplicações implementadas ou alojas na *Cloud* como o tempo de resposta e o estado dos componentes da aplicação;
- **Log Analytics** - uma análise mais profunda sob forma de logs das operações feitas sobre os servidores como *patches* e mudanças nas bases de dados;
- **Security Center** - estado de saúde das VMs, Redes, SQL e Dados, assim como recomendações a tomar.

3.2.2 *Open Stack*

O *OpenStack* é um *framework* de *cloud* privada IaaS que se tem tornado bastante popular nos últimos anos. Surgiu em 2010, de trabalho conjunto da *Rackspace* e da National Aeronautics and Space Administration (NASA), que começaram por procurar uniformizar a gestão das múltiplas VMs hospedadas nos mais variados hipervisores, a gestão dos recursos da infraestrutura, e ainda oferecer uma tecnologia para armazenar eficientemente

objetos não estruturados. Entre os hipervisores suportados pela tecnologia *OpenStack* destacam-se o Kernel-based Virtual Machine (KVM) *(Linux)* e *Xen (Citrix)*, para os quais existem "*drivers*"gratuitos, e o *ESXi (VMware)* para o qual é necessário adquirir uma licença.

A arquitetura é modular e flexível, pois integra vários projetos/componentes independentes. A implementação é feita apenas sobre os componentes que são necessários, sendo eles: [18]

- ***Nova***: Primeiro a ser desenvolvido tem como objetivo gerir os hypervisores, assim como o ciclo de vida das instâncias de *cloud computing*.
- ***Swift***: É um software, que usa chamadas Application Programming Interface (API), para guardar e ler grandes volumes de dados não estruturados. Construído com o objetivo de permitir a escalabilidade dos volumes geridos, assim como otimizar o acesso e consistência pelo conjunto de dados não estruturados. É equiparado ao serviço *Simple Storage Service (S3)* da *Amazon*.
- ***Cinder***: Este componente permite a criação e administração do armazenamento de blocos de dados. Em outras palavras, *Cinder* permite ao utilizador utilizar volumes adicionais.
- ***Glance***: Responsável pelo armazenamento e gestão das imagens dos sistemas operativos no *OpenStack*.
- ***Keystone***: Como o nome indica, gere a segurança da autenticação de serviços e utilizadores. Ou seja, o *Keystone* autoriza outros componentes do *OpenStack* a comunicar, além de gerir as permissões de cada utilizador na dentro da nuvem.
- ***Neutron***: Inclui um conjunto de APIs, *plug-ins* e *softwares* que, basicamente, garantem a transparência na comunicação entre dispositivos e tecnologias dentro de ambientes IaaS.

3.3 Híbrida

Um exemplo de monitorização híbrida, é a integração do SCOM com o *Azure Log Analylitcs*. Ou seja, ter um serviço no SCOM que permite enviar dados para a *Azure Cloud* para serem analisados e tratados. Ideal para um ambiente híbrido pois elimina a necessidade de duas interfaces ou dois sítios diferentes para monitorizar o estado da infraestrutura quer local quer na nuvem. Uma das vantagens da interligação do SCOM com a *Azure* é o facto do tráfego de *upload* ser gratuito.

O cenário contrário também é possível, ter uma infraestrutura em nuvem e monitorizar os agentes nas máquinas *on-premises* com o uso do *Azure Monitor*. Com a entrada do *Windows Server 2019*, funcionalidades como *Windows Admin Center* permitem, para cada umas das máquinas *on-premises*, ativar a integração do *Azure Update Management* (em *Updates tool*), que irá instalar o agente Microsoft Monitoring Agent (MMA) e irá transmitir os dados recolhidos para uma diretoria *Log Analytics* no *Azure Monitor*.

Capítulo 4

Gestão de Serviços *IT*

Antes de explorar como é feita a gestão de serviços IT, é necessário conhecer o que é a ITIL e o que representa. ITIL é uma *framework* de gestão de serviços IT (Information Technology Service Management (ITSM)) muito popular com principal objetivo de proporcionar um gestão uniformizada dos diferentes serviços com um conjunto de boas práticas para atingir uma experiência consistente alinhadas com as necessidades do negócio.

Atualmente, a ITIL vai na versão 4, surgiu na década de 80 na Grã-Bretanha nos centros de informação governamentais. Desde então, foi adaptado e revisto para uso nas diferentes indústrias e organizações, tornando-se um standard na gestão de serviços IT.

Figura 4.1: *ITIL Service Life Cycle*.

Na criação/distribuição de serviços críticos é necessário uma base por suportar e facilitar os processos, procedimentos e tarefas IT para garantir a disponibilidade dos recursos para o desenvolvimento do negócio, assim como, medir e avaliar o tempo gasto na resolução dos processos. O ciclo de vida dos serviços IT, segundo o standard ITIL dividem-se em cinco fases, ver Figura 4.1.

Segundo a ITIL, um serviço é um valor acrescentado fornecido aos clientes para os ajudarem os objetivos desejados para o desenvolvimento do negócio, sem ter que assumir alguns riscos ou custos colaterais associados com este serviço. Ou seja, os clientes só

precisam de se preocupar em consumir o serviço e não como ele funciona por trás.

- **Estratégias de Serviço** – é o planeamento dos serviços com um conjunto de regras que o provedor de serviços e o cliente necessitam de precaver para suportar o negócio de acordo com a estratégia da sua organização.
- **Desenho de Serviço** – a fase mais importante correspondente ao desenho dos processos e serviços descritos na fase anterior, onde, por exemplo, é definido a entrega dos serviços - nível, disponibilidade e capacidade.
- **Transição de Serviço** – refere-se a transição/implementação dos serviços num ambiente de produção. É expectável a ocorrência de problemas de rápida resolução com o mínimo de impacto para o utilizador, podendo deste modo os serviços serem mudados e cancelados baseados em testes e adaptações.
- **Operação de Serviço** – contém um conjunto de instruções às equipas de operações para tratarem das atividades de suporte e entrega do serviço ao utilizador.
- **Melhoria Contínua de Serviço** – nesta fase é verificado se os objetivos foram cumpridos e o que se pode fazer para melhorar os serviços de acordo baseado em medidores de desempenho e documentação de modo a tornar o serviço mais eficiente.

4.1 *Microsoft System Center Service Manager*

O Microsoft System Center Service Manager (SCSM) [19] é uma das muitas ferramentas ITSM, construída seguindo as regras ITIL e princípios operacionais Managed Object Format (MOF), popular na distribuição de soluções centralizadas de tratamento de pedidos, gestão de incidentes, problemas, etc. Integra funcionalidades do Microsoft System Center Configuration Manager (SCCM), Microsoft System Center Operations Manager (SCOM) e AD. Oferecer ao utilizador final ferramentas *self-service* e de submissão de pedidos é um aspecto crucial para simplificar e melhorar as operações de *service desk* IT. Normalizar, automatizar e monitorizar o processo de suporte oferecido pela organização torna-se mais simples, registando os fluxos de processos em *Workflows* visuais.

Os pedidos são constituídos por um sequência de atividades customizáveis, com maior potencial de automação se efetuados no Microsoft System Center Orchestrator (SCORCH), podendo estas (as atividades) serem automáticas, manuais, paralelas ou em série (ver Figura 4.2).

A arquitectura do SCSM divide-se em seis componentes (ver Figura 4.3) :

- **Management Server** é a peça central que permite correr *workflows* e controla os acessos a partir da consola e outros conectores;
- **Service Manager DB** é o sistema de informação com os itens de configuração, registo de incidentes, mudança de pedidos, configurações do ambiente, assim como os atributos principais guardados nos MPs (estes MPs seguem a mesma lógica que os MPs do SCOM apesar de guardarem objetos diferentes);

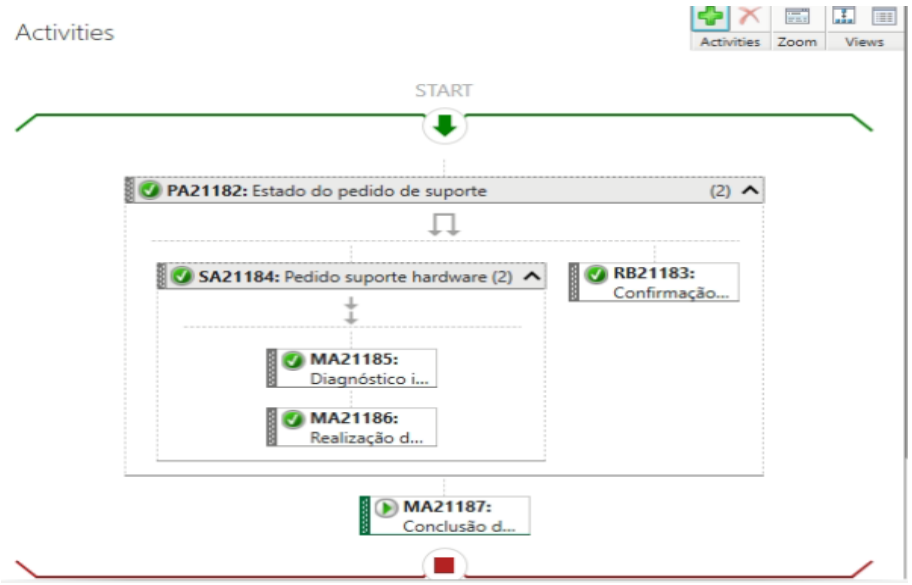


Figura 4.2: Exemplo de um *Workflow*.

- **DataWarehouse DB** guarda dados à longo prazo (que têm pouca utilização e pouco acesso) para efeitos de estatística e relatórios personalizados;
- **Data Warehouse Management Server** controla os acessos à base de dados para a criação de relatórios e notificações sobre as mesmas;
- **Console** é uma Graphical User Interface (GUI), para o ambiente do SCSM usado pelas equipas de *help desk*, resolução de incidentes e administradores de sistemas para tratar dos pedidos, incidentes e outros *workloads*;
- **Self-service Portal** é uma maneira simples e uniforme dos utilizadores autonomamente procurarem soluções, como a reposição da palavra-passe, normalizando e automatizando os pedidos de suporte.

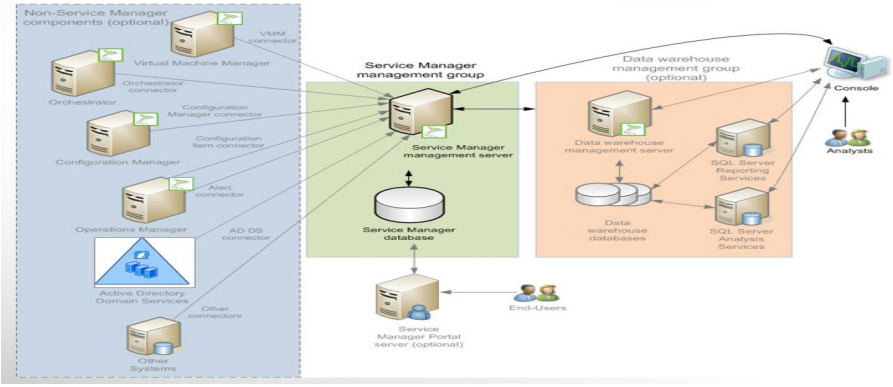
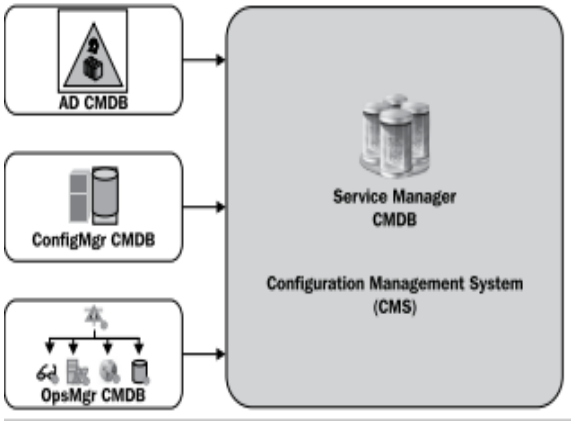


Figura 4.3: Arquitectura do *SCSM*. [20]

4.1.1 Administração

O SCSM pode interligar-se várias fonte de dados ou Configuration Management Databases (CMDBs), do modo a criar a sua própria CMDB com base nelas. A criação de um processo CMS, segundo a Figura 4.4, com pelo menos uma CMDB é feita no módulo de Administração através dos diversos conetores, ver Figura 4.5. Os principais conetores podem ser:

- Active Directory (AD);
- System Center Configuration Manager (ConfigMgr);
- System Center Operations Manager (OpsMgr);
- System Center Operations Orchestrator (Orch).



A criação de um processo dinâmico, isto é, um CMS que respeite as regras ITIL, na medida em que, o refrescamento dos dados obtidos das outras CMDBs é feita automaticamente e diretamente sobre a fonte evitando inconsciências de sincronização. Este módulo ainda contém informação sobre os MPs; modelos de notificações; contas de segurança e *workflows*.

Figura 4.4: Exemplo *CMS*

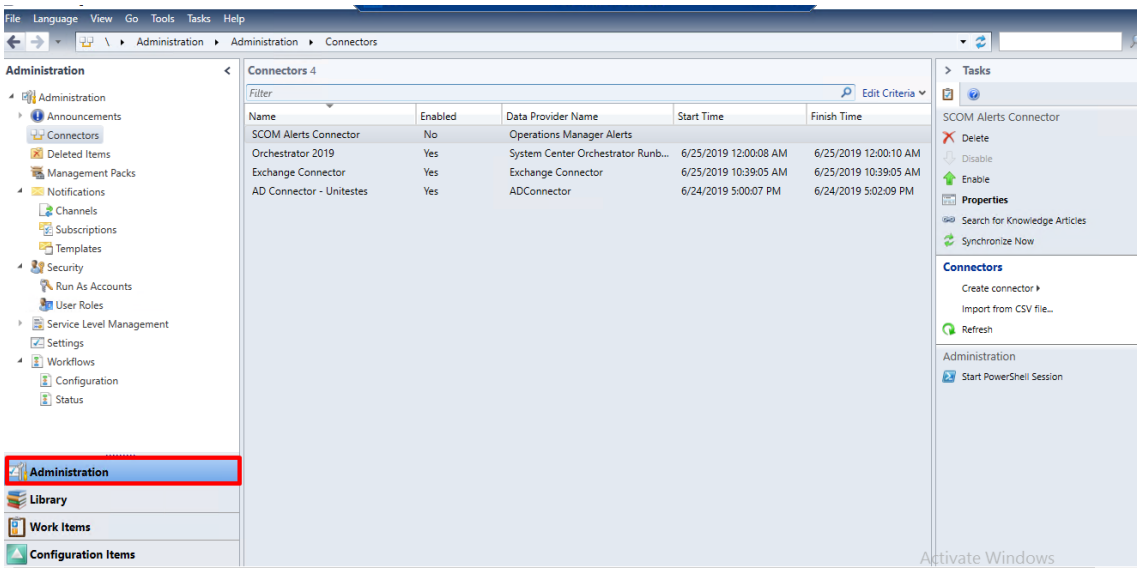


Figura 4.5: Vista de Administração.

4.1.2 Biblioteca

O módulo *Library*, ver Figura 4.6, contém a informação acerca dos *runbooks* sincronizados com o SCORCH, das listas de categorias/sub-categorias disponibilizados nos *request offerings*, que por sua vez fazem parte dos *service offerings*. Um *service offering* pode conter vários *request offerings*, em que para cada um, existe um *template* correspondente com os mapeamentos corretos do *input* do utilizador ou campos já pré-configurados. Os templates podem ser customizados usando uma ferramenta denominada *Service Manager Authoring Tool* para conter informação personalizada adequada para cada tipo de negócio; e ainda contêm a definição do fluxo de atividades a aprovar/recusar quando um pedido for atribuído ao técnico.

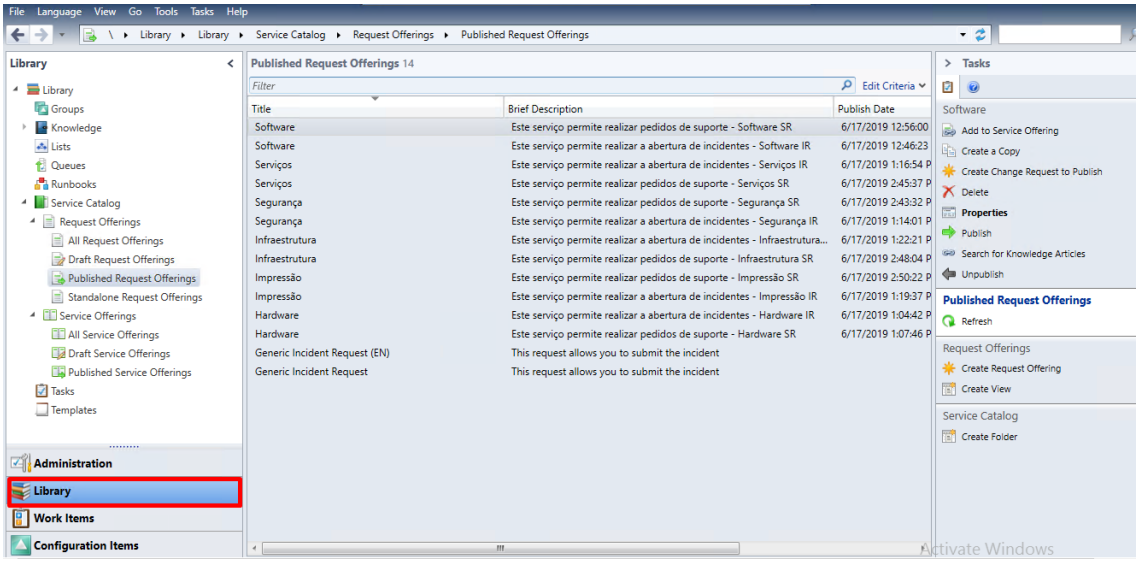


Figura 4.6: Vista de Biblioteca.

4.1.3 Itens de Trabalho

Este módulo apresenta os vários pedidos/incidentes que chegam à consola para serem tratados. É possível definir vistas especiais com base nas regras de segurança, por exemplo, reduzir a vista para equipas que só tratam de um tipo de pedido.

4.1.4 Itens de Configuração

Neste módulo, ver Figura 4.7, estão apresentados os vários *assets* recolhidos pelos conetores: *software*, *hardware*, actualizações, utilizadores, etc, dependendo do tipo de conetor.

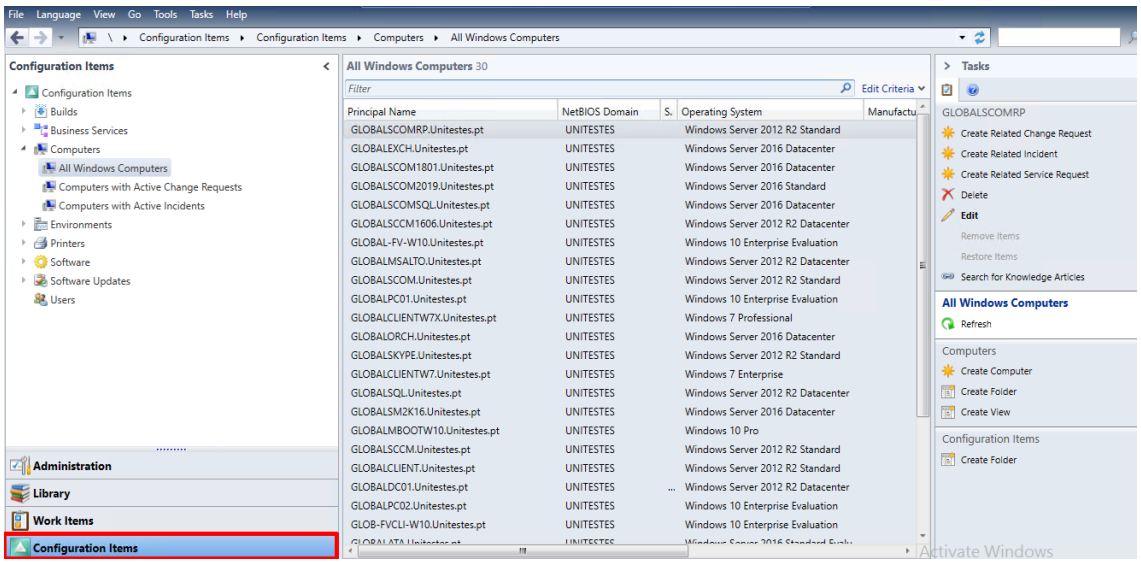


Figura 4.7: Vista de Itens de Configuração.

4.2 *Cireson Portal*

A consola Web é o ponto de interação primário dos utilizadores com a ferramenta de *help-desk*. A consola web do SCSM tem-se relevado ser limitada e rústica quando comparada com as outras do mesmo domínio, nomeadamente quanto às opções de customização, interação e *learning curve* com o utilizador.

Por esforço da comunidade foi desenvolvido um portal *Web* para complemento/substituição do portal *Web* do *Service Manager*, o *Cireson Portal*. A arquitetura do *Cireson Portal* é constituída pelo portal *web*, por uma base de dados própria (opcional) que é sincronizada com a base de dados operacional do SCSM. A razão por não se usar diretamente a base de dados operacional do SCSM é porque o portal da *Cireson* foi desenvolvido para ser mais ágil a efetuar as *queries* sobre a sua própria base de dados, até para guardar as várias versões do portal. Uma vantagem acrescida, para as equipas técnicas, é ser possível operar diretamente no portal *web*, permitindo todas as funcionalidades da consola do SCSM, substituindo deste modo a complexidade que é trabalhar/instalar a consola do SCSM. O portal da *Cireson*, grátis na versão base, oferece informação sumariada em *dashboards* e tempo real do estado dos fluxos de execução.

4.3 *Microsoft System Center Configuration Manager*

Conhecido inicialmente como Systems Management Server (SMS) tinha como objetivo gerir os vários tipos de sistemas numa infraestrutura. Só em 2007 foi renomeado como membro da família *Software Center*. As principais funcionalidades do SCCM são a inventariação de *hardware/software*; a distribuição rápida do serviço às sub-unidades da organização como pacotes de aplicações, atualizações e sistemas operativos; gere a componente de segurança e *compliance* nas máquinas; mede a utilização de *software*; permite

o controlo remoto nas máquinas e a definição de janelas de manutenção, e qualquer configuração a nível de política, *power management*, etc. Um dos *roles* mais impactantes que o SCCM pode assumir é a gestão total dos diferentes tipos de sistemas seja ela *Windows*, *Unix/Linux*, *MacOS*, assim como a gestão dos dispositivos móveis quando integrado com o *Microsoft Intune*, permitindo uma total gestão da infraestrutura híbrida; assume também o *role* Windows Server Update Services (WSUS) para aplicação de *patches* e atualizações; e *Endpoint Protection* para gestão da segurança dos dispositivos.

Aquando ao *deploy* do SCCM deve-se ter em conta a dimensão da infraestrutura. A arquitetura requer ser dimensionada correspondente de modo a responder as necessidades do negócio. A arquitetura do SCCM usando os diversos componentes é bastante complexa, os principais componentes [21] são, ver figura 4.8:

- **Central Administration Site (CAS)** - É a camada abstracional que gere vários *sites* primários a partir de um único ponto principal;
- **Primary Site** - o servidor que pode gerir até 170 mil dispositivos (clientes e servidores) com uma base de dados própria;
- **Secondary Site** - é um filho (nó) do *Primary Site* que alberga até 15 mil dispositivos. O *Secondary Site* permite ser implementado, por exemplo, numa filial do *Primary Site* que é difícil gerir diretamente devido a latência e tráfego na rede. Deste modo, o *Secondary Site* tem uma replicação e sincronização para a sua própria base de dados. Este método torna-se consumidor de recursos e custos adicionais evitáveis com a utilização de um Distribution Point (DP).
- **DP** é um servidor de partilha/distribuição de ficheiros, com suporte até 5 mil dispositivos. Com a vantagem de, por exemplo, na distribuição de pacotes, aplicações ou imagens de OS, é feita somente uma vez para o DP do *Primary Site*, e o DP faz a distribuição para a sua sub-rede, poupando a largura de banda.
- O SCCM é adequado para qualquer tamanho de infraestrutura, mas para dimensões com até 725 mil clientes é necessário um **CAS** que tem a única função de gerir e distribuir os dados entre os vários *sites* primários e secundários, e manter a sincronização entre eles.
- **Management Point** providencia relatórios e *dashboards* personalizados acerca do ambiente SCCM com objetivo de monitorizar o estado de saúde da arquitetura do SCCM, sem necessitar de instalar a consola, possibilita receber a informação dos relatórios e *dashboards* via *email* agendados. Suporta até 25 mil clientes.

Para atingir uma correta gestão do SCCM é necessário garantir que os vários *site system roles*, ver tabela 4.1 , instalados no SCCM estão a funcionar corretamente. Para isso, existem um conjunto de tarefas diárias/semanais/mensais a realizar, em que uma solução de monitorização como o SCOM oferece as vantagens de: disponibilidade dos servidores e execução dos serviços principais no servidor central do SCCM e nos DPs, como o *SMS*

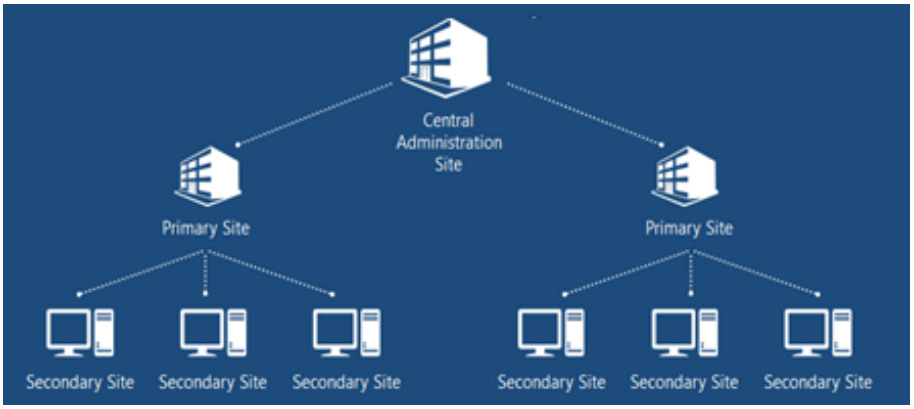


Figura 4.8: Arquitetura *SCCM*. [21]

Agent Host, *SMS_EXECUTIVE*, *SMS_SITE_SQL_BACKUP*, *MSSQLSERVER* e *SQLSERVERAGENT*; monitorização com base em relatórios diários sobre o estado dos discos, RAM e CPU. Configurar o backup do SCCM automático, ou verificar o estado de saúde dos agentes SCCM com base em coleções de máquinas para o efeito, ou configurações dos actualizações específicas providenciados pelo WSUS, assim como o estado do *site* primário e dos seus componentes são outros pontos importantes para uma correta administração da infraestrutura do SCCM que por sua vez gere os restantes servidores/*workstations*. Por sua vez, preferencialmente, é possível executar *queries* na base de dados do servidor primário para obter a mesma informação resumida na consola do SCCM.

A questão que se coloca será que é possível automatizar este processo de manutenção da infraestrutura do SCCM? Ao dividir o problema em sub-problemas e arranjar automatismos isolados, por exemplo a execução de uma *query* SQL, estes (os automatismos) poderão ser integrados numa ferramenta de orquestração (descrita no sub-cap. 5.1.1) para aumento da eficiência e distribuição de tarefas para as equipas responsáveis.

Tabela 4.1: *SCCM Site Components*. [22]

Componente	**Descrição**
Software distribution	*Deployment* remoto em larga escala de aplicações e com relatórios e *dashboards* sobre o progresso da instalação e problemas associados.
Software update point	Permite a distribuição de *Windows Updates*
Operating system deployment	Permite a instalação de sistemas operativos por rede ou *offline* por *boot media*.
Management point	Fornece os serviços aos vários *Boundary Groups* definidos (gama de endereços Internet Protocol (IP)) com as máquinas cliente.
Status reporting	Permite criação de relatórios de sites e clientes.
Email notification	Permite o envio de notificações.
Collection membership evaluation	Mede a sincronização e utilização das coleções de dispositivos e utilizadores.

Capítulo 5

Automação/Orquestração

A automação introduz imensas vantagens na eficiência de execução duma tarefa e diminuição do erro humano. O problema principal associado à automação é a maneira como é desenhada, ou seja, a inexistência de inteligência suficiente quando deparada com situações anormais. Ao falhar pode não dar um *feedback* ou visibilidade adequada ao administrador do sistema podendo causar *automation surprises*[23]. Para isso, é necessário desenhar uma automação mais inteligente e o mais completa possível que assuma a existência de potencias erros.

5.1 *On-premises*

Para sistemas *on-premises*, automação é feita com ferramentas de configuração e gestão que são usadas para automatizar configurações e garantir a conformidade da/com a infraestrutura. Apesar de oferecerem diversas formas de atingir o mesmo resultado – como gerir uma infraestrutura em larga escala com o mínimo de interação humana, cada tecnologia tem as suas vantagens e desvantagens, que iremos seguidamente e de forma sucinta apresentar:

- **Ansible** é um produto *software* escrito em Python que, ao contrário dos produtos concorrentes, não requer um agente instalado: um servidor que tem a localização dos nodes, inicia temporariamente uma conexão remota via Secure Shell (SSH) para comunicar com o nó alvo e efetuar as tarefas de maneira idempotente, ou seja, quando o estado desejado é alcançado os comandos posteriores já não alteram esse estado. [24, 25];
- **Chef** é uma ferramenta escrita em Ruby, uma Domain Specific Language (DSL) para definir *recipes* de configuração que podem ser agrupados em *cookbooks*. As *recipes*

descrevem o estado de um conjunto de recursos: pacotes a serem instalados, ficheiros a serem escritos e serviços a correr. Recomendado tanto para infraestruturas *Cloud*/*on-premises*/híbrida e de tamanho variado. É flexível correndo em modo agente/servidor, em que o agente faz *pull* das configurações, ou em modo Standalone, com uma personalização desejada; [25, 26, 27]

- A *PowerShell Desired State Configuration (DSC)* permite especificar os recursos como um ambiente de um *software* ou serviço que deve ser configurado de forma declarativa para atingir o estado descrito. O Local Configuration Manager (LCM) é o motor responsável por aplicar as configurações aos recursos e periodicamente consultar o estado do sistema, usando *pulls*, para garantir que o estado de configuração é mantido [28, 29]. O método *push* é unidirecional e imediato, pois são especificadas as máquinas alvo [30];
- **SaltStack** é uma plataforma escrita em *Python* que tem as vantagens de ser escalável, modular e extensível. A plataforma suporta tanto o modelo *master/agent* como o modelo descentralizado. Os módulos, que facilmente encaixam, tratam de partes especificas do sistema. Os módulos podem ser categorizados em: *Execution Modules* representam as funções de execução diretas no *Remote Execution Engine*; *State Modules* que mudam a configuração do sistema; *Grains*, responsáveis por detectar informação do sistema e guardá-la em RAM; *Renderer Modules*, usados para renderizar informação para o estado de sistema num formato serializável; *Returners*, que têm a tarefa de gerir as respostas das chamadas de execução remota que provêm de localizações arbitrárias; e *Runners*, aplicações que oferecem funcionalidades adicionais via linha de comandos; [25, 31]
- **Puppet** é uma solução de orquestração, aprovisionamento, configuração e visualização de *reports* construída em *Ruby* [25]. Onde são descritos os recursos do sistema e o estado desejado (paradigma declarativo) com base em *Puppet* ou *Embedded Ruby templates*. A arquitetura usa um modelo *agent/master* em que o agente recolhe as informações de configuração do servidor (*master*), contudo pode ter um modelo descentralizado; [32]
- ***Microsoft System Center Orchestrator (SCORCH)***, ver sub-seção 5.1.1;

Podemos categorizar as ferramentas analisadas, ver tabela 5.1, quanto ao método de configuração, nas quais o servidor escolhe quando enviar as configurações para o destino (*push*), ou nas quais o servidor-alvo pede as configurações (*pull*).
Quanto à abordagem, a declarativa foca-se em declarar o estado final que queremos atingir, deixando para o sistema de suporte a tarefa de como atingir este resultado. A abordagem imperativa faz uso de scripts ou comandos específicos e sequências para mudar a infraestrutura para atingir o estado final desejado. A abordagem inteligente, e mais sofisticada, foca-se no porquê e determina o estado final desejado tendo em consideração o impacto sobre as relações e dependências das várias aplicações (ambiente) da infraestrutura, e só então executa as tarefas para atingir este estado. [33]

Tabela 5.1: Ferramentas *IaC*. [33]

Ferramenta	Método	Abordagem
Ansible	Push	Imperativa
Chef	Pull	Imperativa
Puppet	Pull	Declarativa
SaltStack	Push	Imperativa
DSC	Push/Pull	Declarativa
SCORCH	Push	Imperativa/Declarativa

5.1.1 *Microsoft System Center Orchestrator*

O *Orchestrator* é um componente da família do *Microsoft System Center*. Nas versões iniciais, era um produto independente com uma licença própria e pouca integração com os outros componentes do *System Center*. Com o objetivo de fornecer aos administradores de sistemas a capacidade de controlarem ou orquestrarem um conjunto de operações automatizadas ou *scripts* sob forma de unidades de execução ou *runbooks*, ver Figura 5.1.

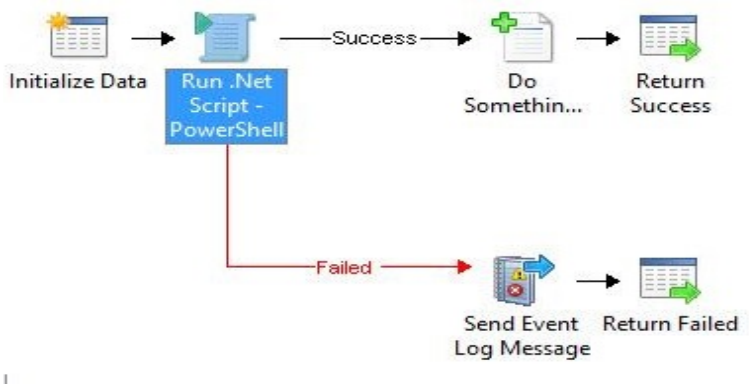


Figura 5.1: Exemplo *Runbook*.

O termo *runbook* surge a partir das instruções e procedimentos para configurar, operar e aplicar num sistema, que eram tradicionalmente compilados num livro; tal documentação não era mais que uma descrição dos pré-requisitos, ações necessárias e cuidados a ter.

A principal vantagem do SCORCH é fornecer uma semântica gráfica e simplificada sob forma de diagramas de atividades os vários processos paralelos a executar para diferentes sistemas. As atividades, ou subunidades que o compõem o *runbook*, têm por base um formato especial dependendo do sistema alvo para que é usada. Ou seja, a instalação a priori de um Integration Pack (IP) de uma plataforma ou aplicação fornece modelos de atividades para serem usadas nesta aplicação.

Uma atividade é representada por um *building block*, um conjunto de passos encapsulados contendo parâmetros de entrada e saída, que é processada segundo uma ordem estabelecida e intercalada com outras atividades (lógica declarativa) sobre o que fazer e quando fazer. As atividades podem ser comparadas, usando uma analogia simples, a

scripts, e podem incluir "comandos".NET, *PowerShell*, SSH, entre outros.

Por norma, os diferentes componentes que fazem parte do SCORCH são instalados no mesmo servidor. A arquitetura[34] do SCORCH (ver Figura 5.2) divide-se em seis partes:

1. O ***Runbook Designer*** serve para criar/editar *runbooks*, validar a execução do *runbook* com o *Runbook Tester* e guardá-los na base de dados;
2. A **base de dados** é uma *Microsoft SQL Server Database* que está no centro da arquitetura, pois contém todos os *runbooks*, informações sobre o estado dos *runbooks*, *logs* e configurações. Toda a informação sobre os *runbooks* é utilizada por outros componentes, numa vertente mais para leitura, ou mais numa vertente para modificações e escrita;
3. O ***Management Server*** é o coração da arquitetura pois faz ligação entre a base de dados e o *Runbook Designer*;
4. Os ***Runbook Servers*** ligam-se diretamente à base de dados para correr os *runbooks* e guardar o *output*;
5. O ***Orchestrator Web Service*** é uma interface que providencia a comunicação com a base de dados das várias aplicações, nomeadamente informações sobre o estado dos *runbook* e operações de iniciar/parar o *runbook*;
6. ***Orchestrator Browser Console*** é uma consola Web, utilizando os mesmos serviços do *Orchestrator Web Service* para comunicar e fazer ações no SCORCH, estas ações podem ser delegadas às diferentes permissões dos diferentes tipos de utilizadores da infraestrutura.

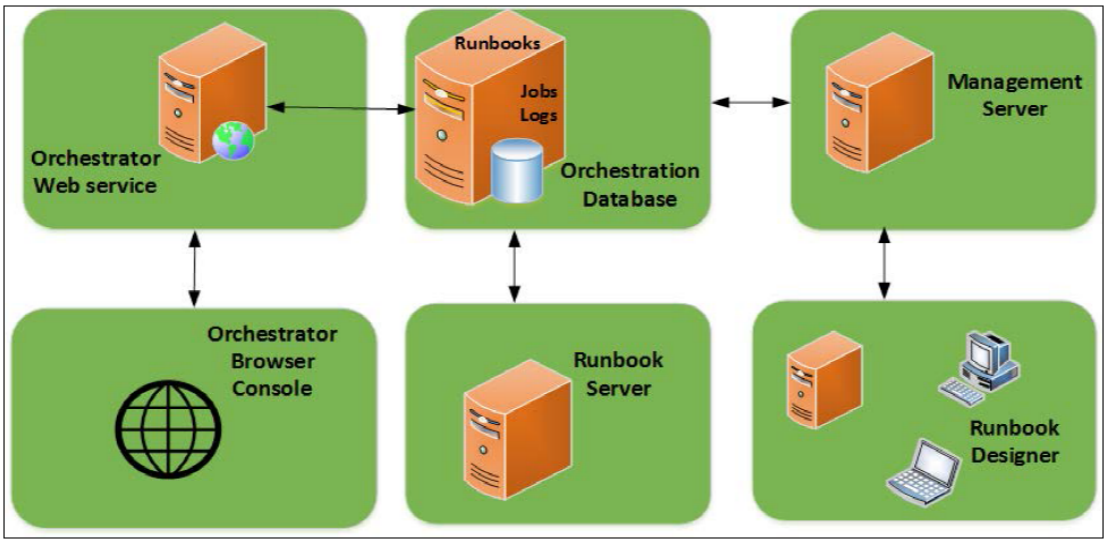


Figura 5.2: Arquitetura do *SCORCH*. [34]

No caso de um ambiente heterogéneo o SCORCH integra ferramentas não *Microsoft* e aumenta a interoperabilidade na infraestrutura; constrói procedimentos consistentes,

compatíveis e documentados; permite automatizar processos segundo as melhores práticas de gestão de pedidos de serviço; reduzir erros e aumentar o tempo de disponibilidade dos serviços, assim como delegar responsabilidades pelos grupos da organização.

5.2 *Cloud*

Apesar de existir cada vez mais uma maior simplificação e banalização ao acesso e utilização da tecnologia, existem tarefas de aprovisionamento e gestão em grande escala que é impensável fazer manualmente uma-a-uma como, por exemplo, a criação de cinquenta máquinas virtuais – esta pode ser um processo demorado e propício a erros.

A existência de um processo de gestão e aprovisionamento de máquinas, tanto Bare-metal como virtuais, usando ficheiros de configuração ao invés de processos manuais, denomina-se IaC. Adequado para *Cloud Computing* (pública e privada) permite reduzir custos, acelerar o processo de aprovisionamento, reduzir o risco de segurança ou diminuir os erros humanos de configuração; além disso permite fazer o "versionamento"e reutilização dos código.

5.2.1 Automação na nuvem *Azure*

A automação do processo de gestão de recursos pode ser feita utilizando duas abordagens: imperativa (*scripting*) ou declarativa (*templates*).

Na *Azure Cloud*, um exemplo de abordagem **Imperativa** é a *Azure* Command-Line Interface (CLI) é uma ferramenta do tipo linha-de-comandos, multi-plataforma, disponível no *Azure portal*, que ajuda a gerir os recursos *Azure* e automatizar os processos com *scripts*. Por exemplo, permite criar uma VM com *Azure* CLI ou adicionar espaço de armazenamento, entre outros. Em alternativa, se preferirmos, podemos utilizar, localmente, *Azure PowerShell* para atingir os mesmos resultados.

Na *Azure Cloud*, Azure Resource Manager (ARM)[35] é uma interface simplificada responsável por criar, gerir e organizar os vários recursos *Cloud*. Permite uma abordagem **Declarativa**, mas não somente, na medida que faz uso de *templates* para criar estes recursos virtuais.
Um *template* ARM é um ficheiro JavaScript Object Notation (JSON) onde é possível definir as propriedades de um conjunto de recursos numa única ação, mas não como criá-los, da mesma forma que os *web browsers* usam os Hyper Text Markup Language (HTML) files que descrevem os elementos que aparecem na página, mas não como os "desenhar"no *browser*. As vantagens do *template* são: aumentam a consistência com o uso da mesma notação; é útil para *deployment* complexos pois existe uma dependência de mapeamentos numa ordem correta; permite estender o *template* com *scripts* para correr dentro da VM; os *templates* podem ser exportados/importados, editados e serem tratados como código; promove a reutilização e organização em módulos. O *deploy* dos ARM *templates* via *PowerShell* foge um bocado à filosofia da abordagem declarativa.

Similarmente ao SCORCH, existe uma ferramenta de automação na *Azure Cloud* chamada *Azure Automation*, que opera sobre o ARM, e segue os mesmos princípios de gestão da infraestrutura e orquestração de processos mas em nuvem. Os componentes da *Azure Automation* sucintamente explicados são:

- *Automation Account* - é a conta sobre qual a automação ocorre, funcionando como um container dos recursos automatizados;
- *Runbooks* - contem a sequência de passos das atividades automatizadas, podendo os runbooks serem do tipo: *PowerShell*, *Python*, *Graphical*, *Powershell Workflow* ou *Graphical PowerShell Workflow*;
- *Jobs* - uma instância da execução do *runbook* pelos *Azure Automation workers*;
- *Assets* - os recursos globais que podem ser associados aos *runbooks*: módulos, conexões, variáveis, certificados, credenciais ou agendas de execução;
- *Hybrid Worker Groups* - máquinas *on-premises* que têm um agente instalado responsável por buscar os *runbooks* da nuvem e executá-los;
- *DSC (Powershell)* - um *Azure Automation DSC Server* aonde os *DSC Nodes* (*on-premises* ou nuvem) vão buscar configurações.

O processo de criação dos runbooks pode ser feita via Portal, ou localmente à partir de uma consola com *PowerShell*. Imaginando uma tarefa de automação, um ARM *template* guardado na nuvem, construir um *runbook* via *PowerShell* que vai a *Storage Account* onde o *template* está guardado, e o importa na *Azure Automation Account*. Por fim, personalizar o *runbook* com uma agenda ou sobre quais as máquinas a executar, e consequentemente, se necessário, adicionar outras atividades ao *runbook*, dividindo a complexidade do processo por várias fases. Em suma, a diferença entre ARM e *Azure Automation*, é que a primeira permite mecanismos de automação de processos, enquanto que a última permite também orquestração de processos automatizados.

5.3 Automação Híbrida

Das ferramentas de orquestração analisadas com mais profundidade (SCORCH e *Azure Automation*), como já visto, tanto uma como a outra permite uma automação híbrida - SCORCH a correr *scripts PowerShell* com módulos *Azure Cloud*, ou por base de IPs especialmente desenvolvidos para fazer a ponte com a nuvem - ou - *Azure Automation* a correr *runbooks* nos agentes *on-premises* (*Workers*). No fundo, depende da estratégia de cada organização e preferência ou nível de conhecimento no domínio para optar uma em relação à outra. A tendência das empresas terem mais recursos *cloud*, também pela minimização dos custos e vantagens de escalabilidade associadas, faz com que adoptem *Azure Automation* para gerir a infraestrutura.

Capítulo 6

Casos Reais - Implementações Realizadas

A atribuição de um projeto de suporte IT para uma cadeia hoteleira com uma infraestrutura IT distribuída internacionalmente pelos 4 continentes é um dos exemplos adoptados para por em prática os conceitos apreendidos nos capítulos anteriores.

6.1 Necessidades do Cliente

A vasta dispersão da infraestrutura IT composta por por cerca de 400 servidores (*Windows Server 2003/2008R2/2012/2012R2/2016*) e 1600 *workstations* (*Windows 7 e Windows 10*), adicionalmente é composta por três domínios que estão ao mesmo nível e dois deles têm uma relação de confiança entre si. Para dar uma visão da infraestrutura ver figura 6.1. A migração infraestrutural baseia-se em boa parte em virtualização sendo gerido com a ajuda do VMM. Os vários *hosts Hyper-V* localizam-se dispersos pelas 3 zonas principais: Portugal, América do Sul e África do Sul. Contudo existem várias máquinas físicas nomeadamente Windows Server 2003 vitais para o negócio que continuam a operar.

Analisando a figura abaixo, existem 4 servidores órfãos que requerem intervenção pois verifica-se que não contêm qualquer replicação para a estrutura de Domain Controllers (DCs) geral. A replicação para África do Sul é feita somente num sentido. Além disso, existem vários pontos de falha, por exemplo, na Argentina, Bahia (Brasil), Rio de Janeiro (Brasil), Venezuela, Madeira (Portugal), Reino Unido, Alemanha e Moçambique. De modo a tolerar a falha de um DC não inviabilizar totalmente a replicação para determinado país. Para isso, analisar e equilibrar os recursos atribuídos a esses DCs secundários.

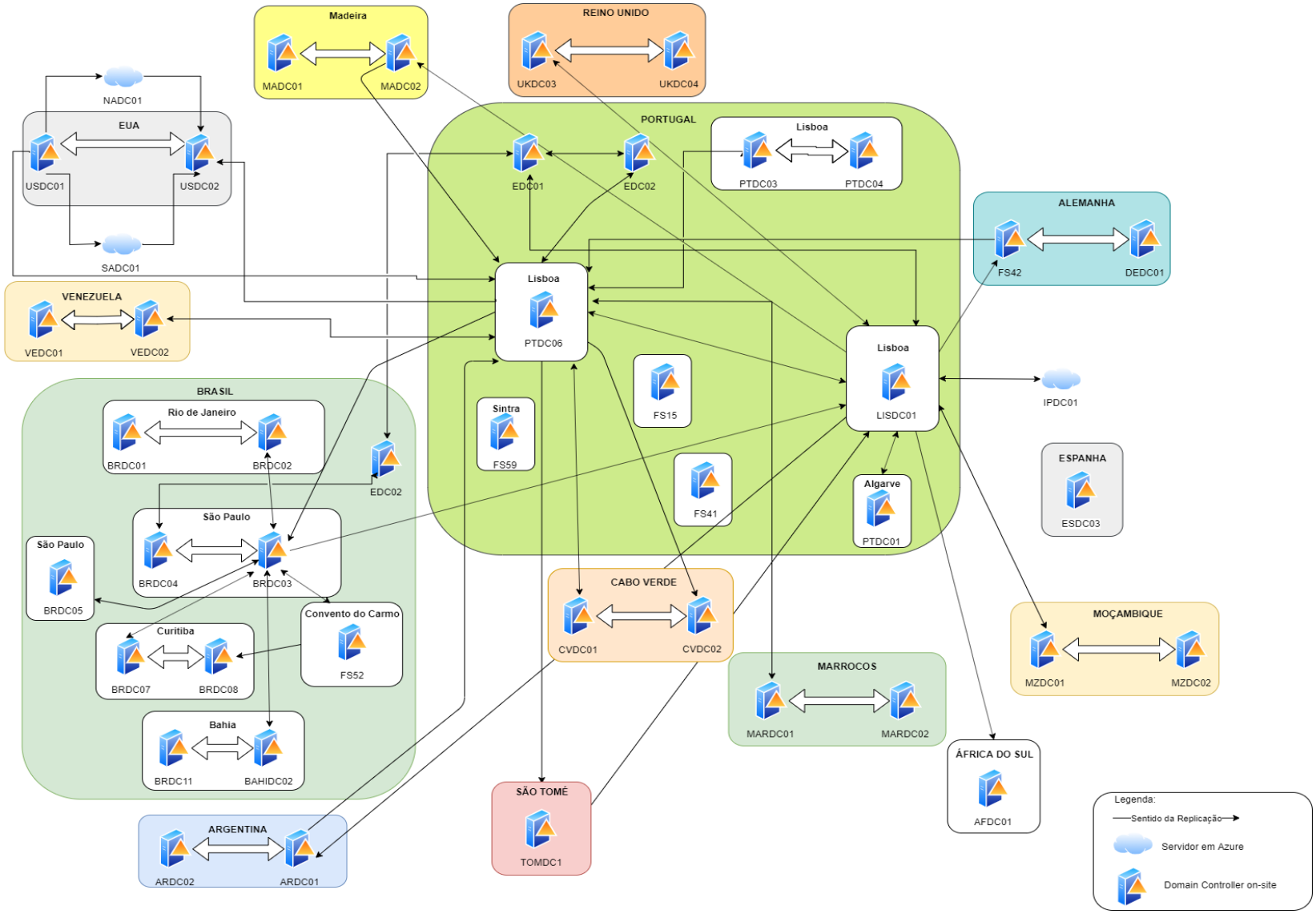


Figura 6.1: Organização da Replicação na *AD*.

Em primeiro ponto o cliente tinha a necessidade de substituir a solução de monitorização que usavam por já se encontrar desactualizada e não ir ao encontro das suas necessidades. Além da funcionalidade base com a vista de alertas críticos que são gerados pelos monitores e regras, queriam ter uma vista por cada país sob forma de *dashboards* do estado da infraestrutura quanto ao espaço em disco, quanto a utilização de memoria RAM e utilização do CPU em cada um dos servidores. Adicionalmente, gostariam de receber semanalmente relatórios com a informação acima referida. Em complemento gostariam de monitorizar as mudanças efetuadas sobre a AD a nível de utilizadores, grupos, máquinas, Organizational Units (OUs), etc para efeitos de auditoria.

Em segundo ponto, havia a necessidade de fazer a manutenção do Microsoft System Center Configuration Manager (SCCM), aplicar e garantir as boas práticas, nomeadamente a grande quantidade de agentes *unhealthy*, e alguns componentes do SCCM apresentavam erros críticos que foram sendo desprezados com o passar do tempo. Nomeadamente:

- Cerca de 200 servidores não recebiam antivírus *definition updates* há mais de 300 dias ou *critical updates* para o OS;
- Cerca de 90 agentes (servidores) SCCM apresentavam um estado não saudável e consequentemente falha na gestão dessas máquinas;
- Infraestrutura de SCCM com falta de manutenção à nível de espaço em disco ocupado para pacotes, mau dimensionamento do disco de backups, coleções desactualizadas ou repetidas com má gestão e organização;
- Garantir e atualizar o processo de *deploy* de imagem de sistemas operativos *Windows 10* para novas sub-redes ou *boundaries*. Vários problemas nos DPs que careciam de atenção e intervenção.

Em terceiro ponto para facilitar o processo e aumentar a eficácia da resolução dos pedidos de suporte que chegavam à equipa de 1ª linha com base na ferramenta ITSM, é necessário incluir alguns automatismos para agilizar o processo. Com o maior enfoque nos pedidos de serviço sobre alterações das propriedades do utilizador na AD ou criação/alteração de propriedades nas caixas de correio *Office 365*. Contudo, os automatismos devem incluir uma aprovação/refutação por parte de um nível de intervenção superior antes da execução do mesmo.

6.2 Implementação de um Sistema de Monitorização Central

Por preferência do cliente escolheu-se Microsoft System Center Operations Manager (SCOM) como sistema de monitorização para a infraestrutura IT. Inicialmente foi feita a documentação da AD com o uso de *script PowerShell* da comunidade [36], onde foram detectados 40 DCs e cerca de 280 *subnets*.

O plano de trabalhos segue a seguinte ordem:

1. Desenho da arquitetura, conforme figura 6.2;

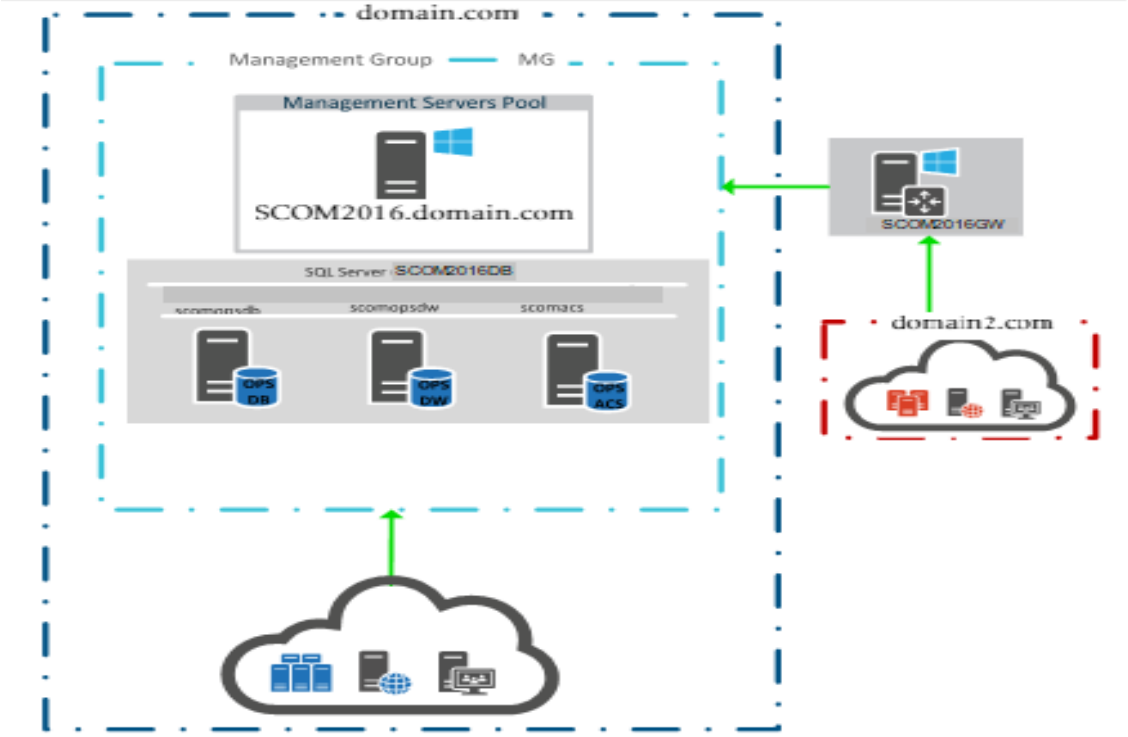


Figura 6.2: Arquitetura *SCOM* do Cliente.

2. Pré-requsitos de software: *.NET Framework 3.5*, *SQL Client*, *ReportViewer*, *SQL Common Language Runtime (CLR) Types*, *Open Database Connectivity (ODBC) 13*.

3. Instalação conforme a arquitectura de:

   a) 3 instâncias de *SQL Server*:

      i. A base de dados operacional que é suposto ser muito eficaz;

      ii. A base de dados com histórico dos itens a longo prazo;

      iii. A base de dados para guardar um conjunto de relatórios integrados ou criar relatórios personalizados com a ajuda do Report Builder.

   b) 1 *SCOM Management Server*;

   c) *Reporting Service*;

   d) *WebConsole* com Secure Sockets Layer (SSL);

   e) *SQL Management Studio 2017*.

4. Registar os Service Principal Names (SPNs):

```
setspn -S MSOMSdkSvc/SCOM2016 domain\SA_OMDAS
setspn -S MSOMSdkSvc/SCOM2016.domain.com domain\SA_OMDAS
```

5. Aplicar o *Update Rollup 6 & 7* para *SCOM 2016*;

6. Configurar o período de retenção da *OperationsManagerDB & Datawarehouse*;

```
dwdatarp.exe -s SCOMDB\SCOMOPS
  -d OpsMgrDB -ds "Performance data set-a "Hourly aggregations-m 60
  -d OpsMgrDB -ds "Performance data set-a "Daily aggregations-m 180
  -d OpsMgrDB -ds "Alert data set-a "Raw data-m 80
  -d OpsMgrDB -ds "Event data set-a "Raw Data-m 10
  -d OpsMgrDB -ds "State data set-a "Raw data-m 60
  -d OpsMgrDB -ds "State data set-a "Hourly aggregations-m 60
  -d OpsMgrDB -ds "State data set-a "Daily aggregations-m 90
```

7. Importar os MPs para:

   a) *Windows Server Operating System (2003,2008,2012,2016);*

   b) *Windows Server Cluster Disks (2008,2012,2016);*

   c) *Active Directory (2008,2012,2016);*

   d) *Hyper-v (2008,2012,2016);*

   e) *Exchange 2013;*

   f) *Microsoft Forefront Threat Management Gateway (TMG);*

   g) *Microsoft System Center Data Protection Manager (DPM);*

   h) *Office 365;*

   i) *BackupExec.*

8. Instalar os 400 agentes SCOM (via *script*):

```
msiexec.exe .\MOMAgent.msi /i /qn NOAPM=1 USE_SETTINGS_FROM_AD=0
MANAGEMENT_GROUP=MG
MANAGEMENT_SERVER_DNS=SCOM2016DB.domain.com
MANAGEMENT_SERVER_AD_NAME=SCOM2016.domain.com
ACTIONS_USE_COMPUTER_ACCOUNT=1
USE_MANUALLY_SPECIFIED_SETTINGS=1 AcceptEndUserLicenseAgreement=1
```

9. Aplicar um *patch* de correção nos DCs:

```
'C:\Program Files\Microsoft Monitoring Agent\Agent \HSLockdown.exe'
  /A 'NT AUTHORITY\SYSTEM'
net stop HealthService
net start HealthService
OOMADs.msi
```

10. Criação de relatórios SCOM com, por exemplo, um *script PowerShell* que é executado diariamente através de uma *Schedule Task* do *Windows*, sob as credenciais da conta de serviço com privilégios de leitura na base de dados *OperationsManagerDW*. O relatório é enviado por *email* (configurado no *script*) com o excerto abaixo apresentado.

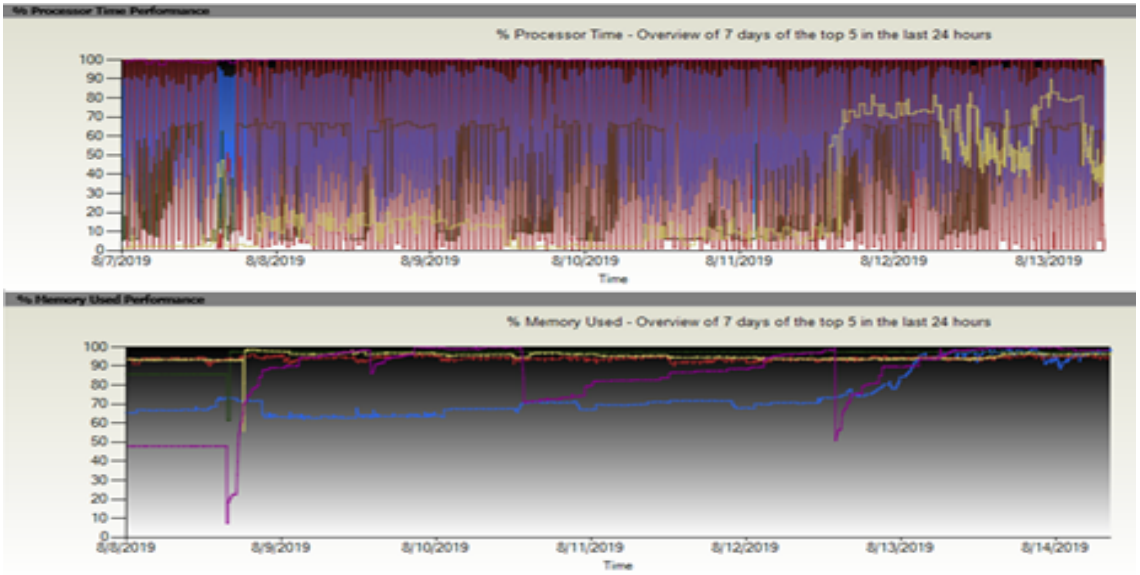


Figura 6.3: Exemplo relatório *SCOM* via *PowerShell*.

11. Criar Subscrições para alertas críticos;

12. Criar *Dashboards* personalizados;

13. Afinação de monitores e regras com base em *overrides* adequado às necessidades do cliente;

14. Instalar o Audit Collection Service (ACS) (*Collector* e DB):

    a) Importar Relatórios ACS pré-feitos;

    b) Editar a *query* com o filtro de eventos a coletar.

```
adtadmin /setquery /collector:SCOM2016 /query:
"SELECT * FROM AdtsEvent
WHERE NOT (((EventId=528 AND String01='5')
OR (EventId=576 AND (String01='SeChangeNotifyPrivilege'
OR HeaderDomain='NT Authority'))
OR (EventId=538 OR EventId=566 OR EventId=672 OR EventId=680)))"
```

O objetivo inicial é reduzir o número de alertas, corrigindo-os ou estabelecer regras de *override* específicas para determinado objeto ou classe de objetos. Normalizada essa fase re-ativa, o próximo passo é, pro-ativamente, identificar servidores virtuais mal dimensionados, isto é, somente com um processador lógico. Para isso, foi desenvolvido o seguinte *script PowerShell* que ajuda a perceber a integração do *PowerShell* com SCOM à base das instâncias da classe "mãe" *"Windows Computers"*:

```
1 Import-Module OperationsManager
2 $class = get-scomclass -Name Microsoft.Windows.Computer
3 $serverOSes = Get-SCOMClassInstance -class $class
```

```
4  $object = @()
5
6  foreach ($serverOS in $serverOSes) {
7      if(($serverOS.'[Microsoft.Windows.Computer].LogicalProcessors'.
       Value -eq "1") -and ($serverOS.'[Microsoft.Windows.Computer].
       HostServerName'.Value -ne "null")){
8          $object += New-Object -TypeName psobject -Property @{
9              "Name"      = $serverOS.DisplayName
10             "Number of Logical Processors" = $serverOS.'[Microsoft.
       Windows.Computer].LogicalProcessors'
11             "Host Server Name" = $serverOS.'[Microsoft.Windows.
       Computer].HostServerName'.Value
12         }
13     }
14 }
15 $object | Select-Object -Property "Name","Host Server Name","Number of
       Logical Processors"  | Sort-Object -Property "Host Server Name"  |
       Format-Table
```

6.3 Tarefas de Manutenção *System Center Configuration Manager*

Os diferentes componentes que compõem o SCCM necessitam de uma monitorização diária com a aplicação de umas tarefas de manutenção ou de *check-up* do sistema de acordo com as boas práticas. Esta verificação pode ser efetuada com *queries* à base de dados, ver Anexo II.

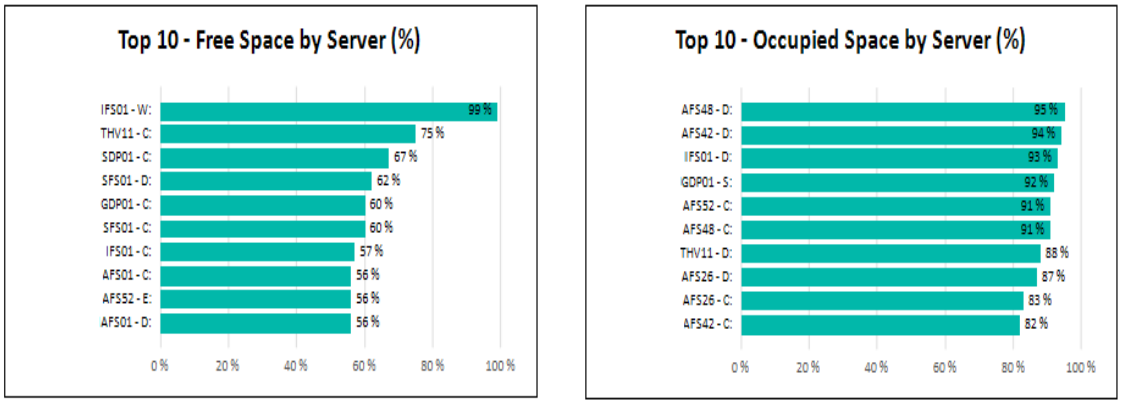
Devido à arquitetura da infraestrutura espalhada pelo mundo, por si, é lógico ter pelo menos um ponto de distribuição localmente a servir os clientes de uma região. Foram identificados 16 DPs e corrigidos os alertas quanto ao estado de saúde destes servidores, entre os quais:

- **Pacotes de aplicações órfãs** - Nos DPs existem um conjunto de pacotes "órfãos", ou seja, não correspondem a nenhum pacote no servidor central que ficaram perdidos. Em média para cada DP, por uso de *scripts*, eliminou-se entre 3 GB - 5 GB de conteúdo "morto"por cada 100 GB ocupados, resolvendo-se alguns problemas de falta de espaço em disco.
- **Pacotes de aplicações inválidos** - Pacotes cuja versão ou conteúdo não foram corretamente transferidos para o DP.
- **Backups** - O *Backup* é automático com uma periodicidade de uma vez por dia e período de retenção muito baixo (só guardava a versão mais recente). Devido a restrições de espaço em disco encurtou-se a periodicidade dos *backups* para 3x por semana e é feito mensalmente um *backup* manual para outra localização.

- **DP com o disco corrompido** - foi identificado um DP que periodicamente a escrita/-leitura nos discos falhava e era necessário o *restart* da máquina. Devido à dimensão que o servidor abrange, todas as 15 redes do Brasil, optou-se por descontinuar o servidor. Inicialmente ponderou-se a possibilidade de ter o servidor na nuvem mas refutou-se a ideia devido ao elevado tráfego gerado, e consequente custo associado, por exemplo só numa instalação de uma imagem do sistema operativo envolve a transferência de 10 GB à 20 GB. A 2ª abordagem num servidor *on-site* consiste instalar o agente SCCM e o role de DP (com o WDS ativo) com objetivo de aprovisionar as tais 15 sub-redes. A mudança da arquitetura do SCCM é um risco que teve que ser devidamente estudado devido a criticidade do mesmo.

- **Criação de coleções** - criaram-se coleções dinâmicas, cerca de 25, à base de *queries* para identificar as características das máquinas quanto ao OS, *software* instalado, marca/modelos, se é máquina virtual, o estado de saúde dos agentes, problemas de disco, *reboot pending*, servidores *Exchange*, SQL, DHCP, DCs, etc.

- **Monitorização do espaço em disco nos DPs** - para este efeito criou-se um grupo no SCOM baseado num relatório semanal especialmente criado para monitorizar os 10 servidores com o espaço em disco mais crítico, ver figura 6.4;



Computer Name (14)	Logical Disk	Disk Size (GB)	Free Space (GB)	Free Space (%)	Date Time	Occupied Space
AFS48	D:	852	43	5 %	8/7/2019 9:56:17 AM	95%
AFS42	D:	649	39	6 %	8/19/2019 6:09:52 PM	94%
IFS01	D:	700	49	7 %	8/19/2019 5:53:53 PM	93%
GDP01	S:	280	22	8 %	8/19/2019 5:32:18 PM	92%
AFS48	C:	78	7	9 %	8/7/2019 9:56:17 AM	91%
AFS52	C:	76	7	9 %	8/19/2019 5:46:32 PM	91%
THV11	D:	1,442	173	12 %	8/19/2019 5:53:44 PM	88%

Figura 6.4: Relatório *SCOM*.

- **Verificação e Validação do processo de atualizações** - Aplicaram-se as boas práticas de transferência e distribuição automática de : *critical updates*, *updates*, *definitions*

updates, para OS, antivírus (Microsoft System Center Endpoint Protection (SCEP)) e produtos relacionados via SCCM. Retificaram-se as políticas aplicadas aos agentes. Relativamente ao período de obtenção das atualizações e ordem de obtenção caso o modo primário falhe. Ficando definido em:

1. SCEP;
2. WSUS - criação de uma excepção automática de autorização para a classe de atualizações pretendidas;
3. *Windows Update* - nos casos da máquina conseguir ligar-se à Internet;
4. *Microsoft Malware Protection Center*;
5. *UNC file shares* - não configurado.

Para o mesmo propósito verificaram-se e corrigiram-se algumas configurações nas Automatic Deployment Rule (ADR). As ADR são regras ou templates para tratamento de determinados tipos de *software* baseado em critérios pré-definidos. Ou seja. tendo um grupo de atualizações críticas para *Windows 10*, automaticamente aprovar e adicionar novas atualizações a esse grupo. [37] Por norma, estas atualizações, que podem chegar aos 50 GB, têm um grande impacto a nível da largura de banda consumida, felizmente, nas configurações das ADR podemos definir os intervalos de tempo sobre o *download* e outro para a instalação dos *updates*. A comunicação entre o agente e o servidor SCCM é limitado pelo Background Intelligent Transfer Service (BITS)[38] que regula a quantidade de recursos da rede utilizada pelo agente;

- ***User Compliance*** - O objetivo era verificar se os *desktops* dos utilizadores eram utilizados pelo proprietário do dispositivo ou *primary user*. Para tal foi desenvolvido o seguinte *script PowerShell* para despistar situações incorretas, e permite-nos dar uma ideia da utilização de *PowerShell* com SCCM:

```
1  #Connect to Primary Site
2  cd 'C:\Program Files (x86)\Microsoft Configuration Manager\
       AdminConsole\bin'
3  Import-Module .\ConfigurationManager.psd1
4  New-PSDrive -Name SiteCode -PSProvider 'AdminUI.PS.Provider\CMSite
       ' -Root 'SCCMServer.domain.pt'
5
6  #Read from file and compare with SCCM Database
7  $file = Import-Excel C:\file.xlsx
8  $file | ForEach-Object {
9      $email = $_.'email'
10     $equipamento = $_.'Equipamento'
11      if($email -ne $null){
12         $var = Get-CMUserDeviceAffinity -UserName $email | foreach
       { $_.ResourceName }
```

```
13          if(($var -eq $equipamento) -and ($equipamento -ne $null)){
14              Write-Host -BackgroundColor green 'User: ' $email ' e
    primary user do equipamento: ' $equipamento
15          }
16          else{
17              Write-Host -BackgroundColor red 'User: ' $email ' nao
    e primary user do equipamento: ' $equipamento ' mas sim do
    equipamento ' $var
18          }
19      }
20  }
```

6.4 Automatismos de tarefas

Para este efeito foram identificados 15 procedimentos chave a serem elaborados para ajudarem as equipas de IT espalhados pelo mundo a seguirem. Os procedimentos foram elaborados para utilizadores iniciantes (via interface gráfica) e para utilizadores avançados (via *scripts PowerShell*). O objetivo é utilizar uma ferramenta de orquestração, que via *scripting*, consegue automatizar processos. Foi escolhido o Microsoft System Center Orchestrator (SCORCH) devido a sua interligação com o SCOM/SCCM/*Office 365*/*Exchange*, entre outros. Devido a extensão do projeto não foi possível implementar a solução a tempo útil de apresentação deste documento. Dado ter sido implementada uma solução de monitorização (SCOM), é expectável introduzir alguns automatismos na resolução de alguns alertas.

Contudo para outro cliente foram implementados os seguintes procedimentos:

1. Criar/Apagar caixa de correio (*Exchange* e *Office 365*);
2. Criar/Apagar caixa de correio (*Exchange* e *Office 365*) e conceder acessos;
3. Criar regras de re-encaminhamento automático de *email*;
4. Migrar uma caixa de correio on-premises para *Office 365* e atribuir licença;
5. Ativar *Litigation Hold* ou o *Archive* numa caixa de correio;
6. Mover utilizadores entre OUs;
7. Criar/Apagar utilizadores na AD;
8. Editar propriedades de utilizadores na AD como grupos e *reset* da palavra-passe;
9. *Auto-close* de incidentes e pedidos de serviço com estado resolvido (ver Anexo III)- Microsoft System Center Service Manager (SCSM);
10. Ciação de uma ativade manual de aprovação por parte do utilizador final - SCSM;

11. Pré-configuração de campos no tratamento nas atividades de um pedido se serviço baseado nas propriedades do técnico/utilizador afetado - SCSM.

Dado os requerimentos do cliente, o processo de resolução dos pedidos de serviço só é possível ou só faz sentido fazer usando uma ferramenta de automação. Não havendo uma comparação ao equivalente se feito manualmente. Para os procedimentos mais regulares para um técnico de 1ª linha, estima-se que o esforço, com a introdução de automatismos, seja reduzida em cerca de 90%. Por exemplo, para o procedimento de criação de uma caixa de correio *Office 365*, um técnico demora em média 15 minutos, enquanto que usando a ferramenta de orquestração SCORCH demora cerca de 2 minutos e 30 segundos. Removendo o tempo de introdução dos parâmetros pelo técnico, a execução do *runbook* demora cerca de 10 segundos.

6.5 Estudos Adicionais

De modo a consciencializar sobre os gastos excessivos numa infraestrutura IT foi feito um estudo sobre o consumo de energia nos postos de trabalho. Além da vantagem principal de reduzir custos diminuindo o consumo de energia total, diminuir as emissões de CO2, otimizar processos, consegue valorizar o produto ou serviço dando uma boa imagem da organização perante as questões ambientais. O plano de trabalhos consiste em:

- Criação de um ou mais planos de energia;
- Monitorizar a compliance/non-compliance das máquinas;
- Relatório sobre o consumo de energia e possível redução de custos;
- Análise dos resultados;
- Implementar numa escala maior o plano de energia.

Foram utilizadas as seguintes métricas como valores fixos nos consumos de energia:

Energy Consumption Constants (kWh):

Desktop computer on	0,2
Laptop computer on	0,045
Desktop monitor on	0,031
Cost of kWh (Portugal 2019)	0,2154 EUR (€)

Figura 6.5: Constantes de consumo energia.

6.5.1 "Balanced" Energy Plan

Foi monitorizada uma máquina representativa do parque aplicando um plano de energia básico, devolvendo os seguintes resultados:

Report Date	Energy consumption (kWh)	Number of Machines Reporting
02/10/2019	2,1011	1
03/10/2019	2,2426	1
04/10/2019	1,9142	1

Figura 6.6: Tabela com relatório de consumo.

Com o seguinte gráfico:

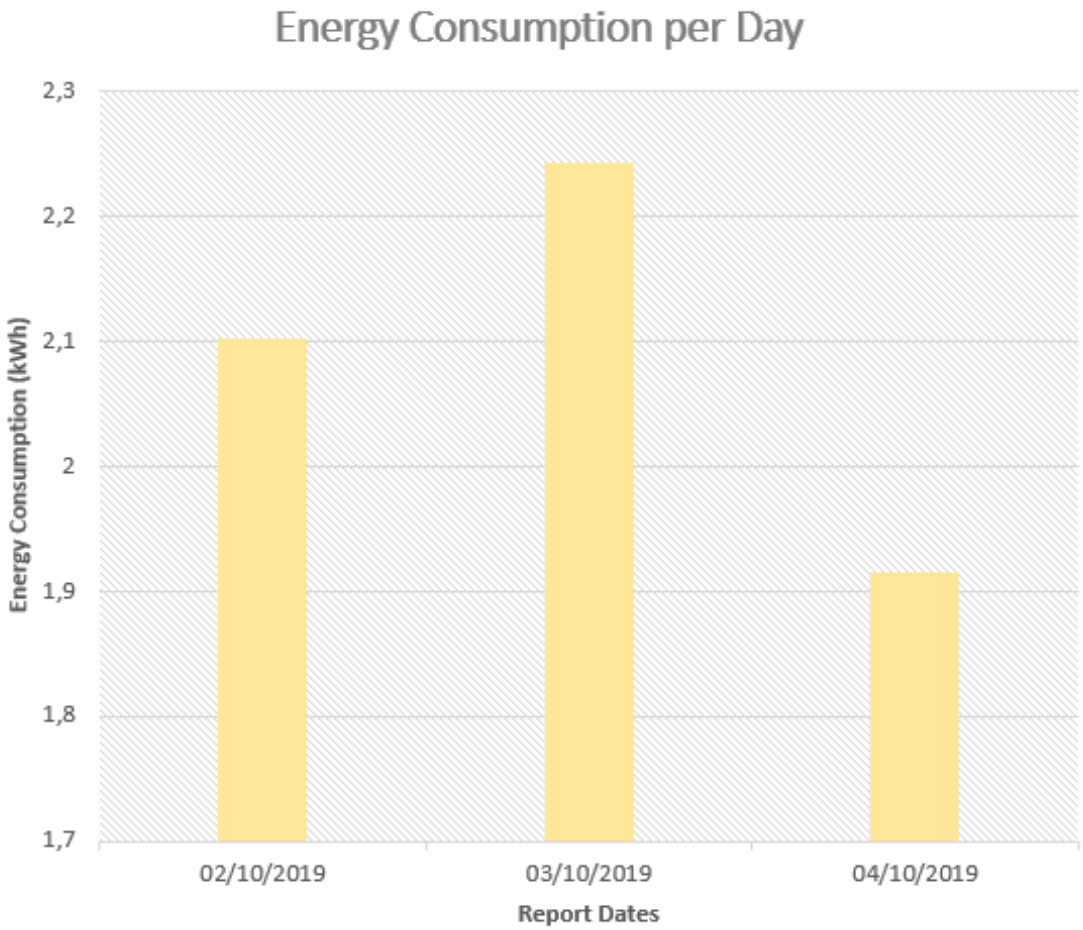


Figura 6.7: Gráfico de consumo de energia com um plano de energia básico.

Para o Cenário 1 com 500 postos de trabalho projetamos o seguinte:

1. Média consumo dia: 2,085 kWh x 500 postos = 1042 kWh (225 €);
2. Média consumo mês: 1042 kWh x 30 dias = 31.260 kWh (6.733 €);
3. Média consumo ano: 31.260 kWh x 12 meses = 375.120 kWh (80.801 €).

6.5.2 "High Performance" Energy Plan

Foi monitorizada uma máquina representativa do parque aplicando um plano de energia de alto desempenho, devolvendo os seguintes resultados:

Report Date	Energy consumption (kWh)	Number of Machines Reporting
07/10/2019	2,1155	1
08/10/2019	3,4995	1
09/10/2019	4,9029	1

Figura 6.8: Tabela com relatório de consumo.

Com o seguinte gráfico:

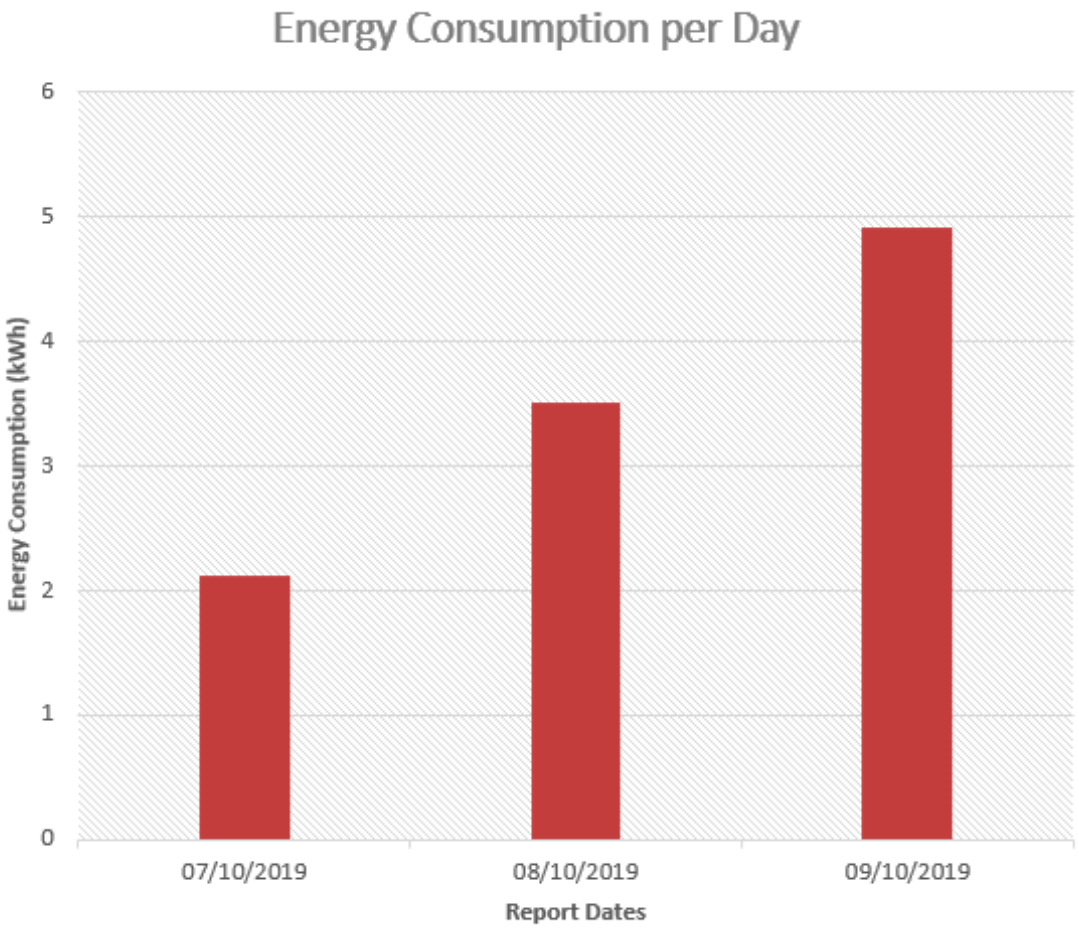


Figura 6.9: Gráfico de consumo de energia com um plano de energia de alto desempenho.

Para o Cenário 2 com 500 postos de trabalho projetamos o seguinte:

1. Média consumo dia: 3,5 kWh x 500 postos = 1750 kWh (377 €);
2. Média consumo mês: 1750 kWh x 30 dias = 52.500 kWh (11.309 €);
3. Média consumo ano: 52.500 kWh x 12 meses = 630.000 kWh (135.702 €).

6.5.3 "Power Saver" Energy Plan

Foi monitorizada uma máquina representativa do parque aplicando um plano de energia baixo desempenho, devolvendo os seguintes resultados:

Report Date	Energy consumption (kWh)	Number of Machines Reporting
07/10/2019	1,5478	1
08/10/2019	1,023	1
09/10/2019	1,0504	1

Figura 6.10: Tabela com relatório de consumo.

Com o seguinte gráfico:

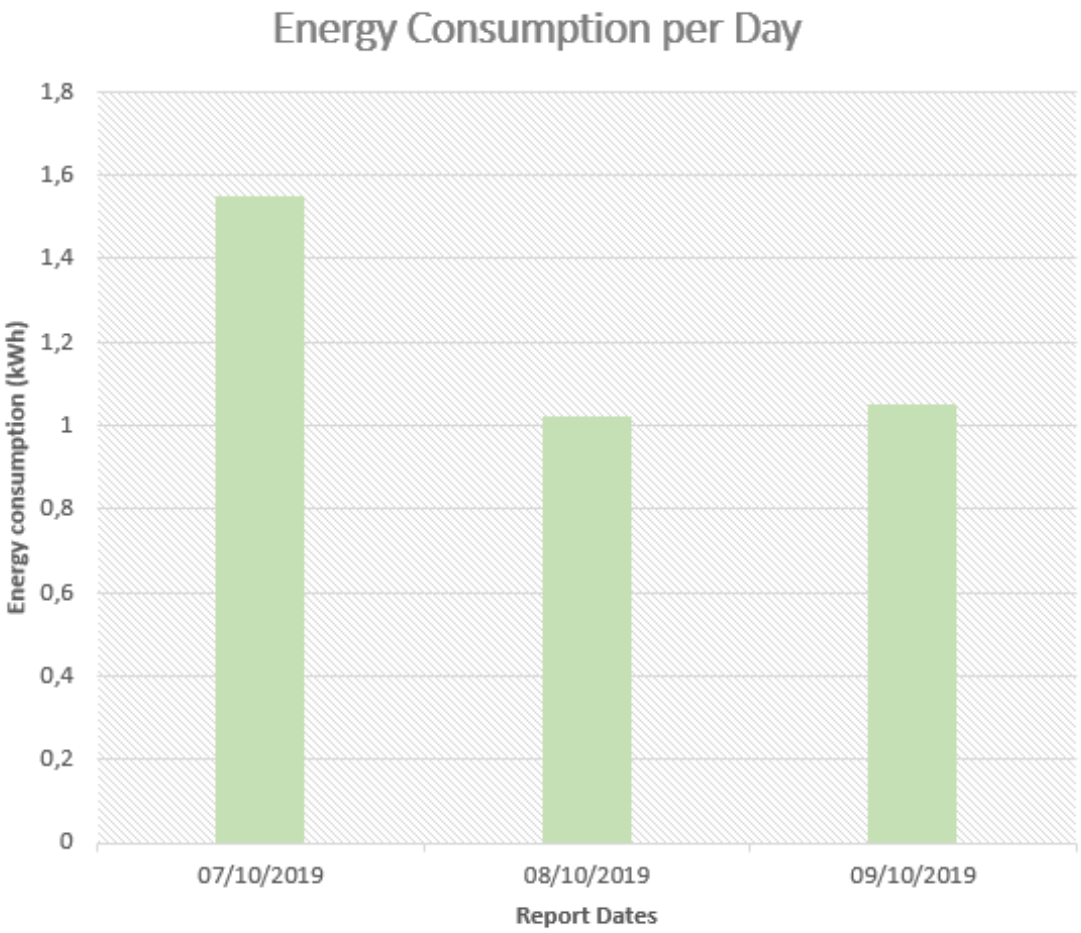


Figura 6.11: Gráfico de consumo de energia com um plano de energia baixo desempenho.

Para o Cenário 3 com 500 postos de trabalho projetamos o seguinte:

1. Média consumo dia: 1,2 kWh x 500 postos = 600 kWh (129 €);
2. Média consumo mês: 600 kWh x 30 dias = 18.000 kWh (3.877 €);
3. Média consumo ano: 18.000 kWh x 12 meses = 216.000 kWh (46.526 €).

6.5.4 Análise dos resultados

Apesar do estudo ter sido realizado com uma amostra de máquinas reduzida e casos práticos de implementação destes planos não ser possível ser feita com o máximo rigor pretendido. Obtivemos os seguintes comparações:

- O Cenário 3 (ideal) apresenta uma poupança de 42% comparando ao Cenário 1;
- O Cenário 3 apresenta uma poupança de 66% comparando ao Cenário 2;
- O Cenário 1 apresenta uma poupança de 41% comparando ao Cenário 2;

A diferença entre os consumos com os diferentes planos é suficientemente considerável para obter uma ideia válida deste estudo e suficientemente apelativa ou interessante para apresentação/reunião com os gestores de uma organização para implementação de um plano de energia, de forma faseada e adaptada à realidade e necessidades da organização.

Esta implementação poderá ser facilmente feita à escala total da organização usando uma ferramenta como o SCCM, conforme figura abaixo:

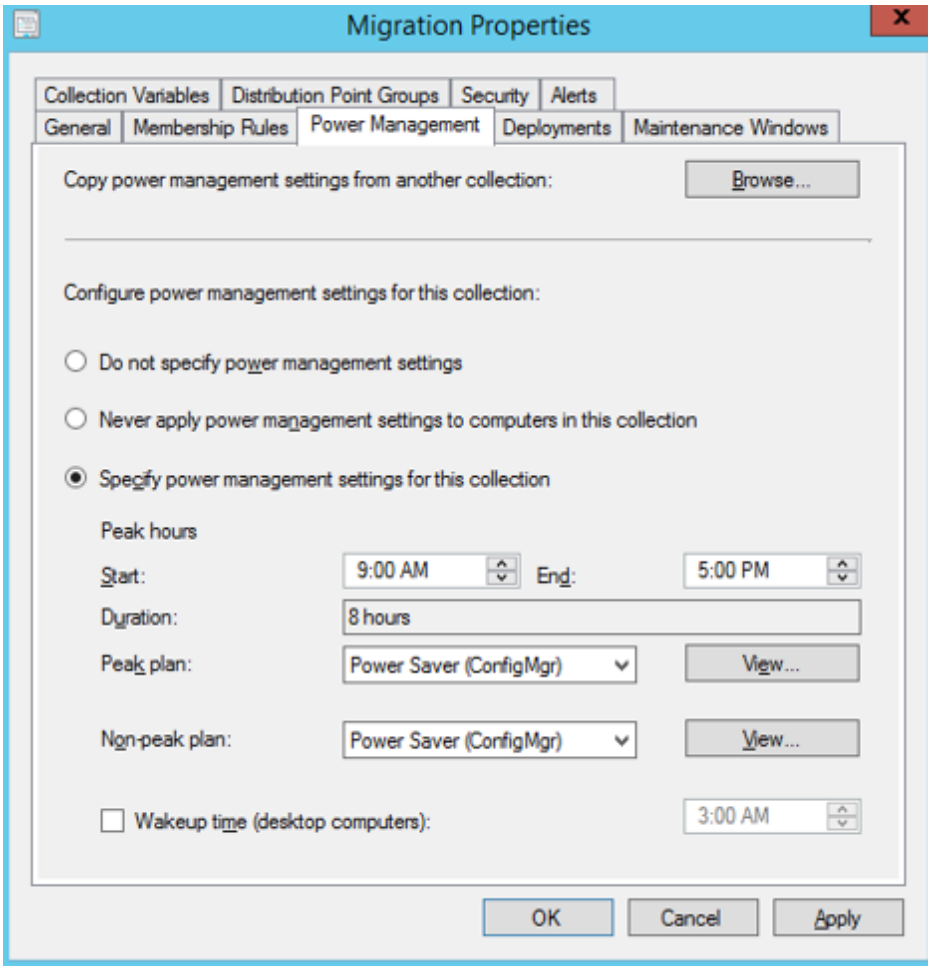


Figura 6.12: Gestão de um plano de energia via SCCM.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

O tema proposto para esta tese é muito ambicioso, existem uma panóplia de temas ou ferramentas sobre as quais podemos ficar perdidos, de facto, gerir uma infraestrutura mesmo com uma equipa IT com experiência, *out-sourced* ou não, é uma tarefa diariamente desafiante. Os desafios com que os administradores de sistemas se deparam são um sinal positivo de que existe uma atualização e um dinamismo nos processos internos que suportam a infraestrutura, que naturalmente requerem um estudo contínuo de melhoramento.

Todo o conhecimento adquirido durante a tese de mestrado foi, em grande parte, posto em prática pelos 4 clientes - organizações com infraestruturas IT bastante complexas. O cenário ideal seria conseguir integrar todas as principais tecnologias estudadas (Microsoft System Center Operations Manager (SCOM), Microsoft System Center Service Manager (SCSM), Microsoft System Center Orchestrator (SCORCH), Microsoft System Center Configuration Manager (SCCM)) num só ambiente, contudo esta cenário só foi possível no laboratório de testes fornecido pela *Unipartner*.

A decisão das tecnologias abordadas estão intrinsecamente ligadas com o "expertise"da empresa nesta área, contudo foram estudadas soluções complementares que permitem ter uma visão das vantagens/desvantagem de uma solução em relação à outra. A ferramenta de ITSM da *Microsoft*, o SCSM, requer uma interface *web* mais elaborada para ultrapassar a competição, contudo o motor do SCSM é extremamente elaborado e robusto.

7.2 Trabalho Futuro

Apesar dos âmbitos dos projetos se relacionarem-se somente com *Microsoft Windows Server* existente nos clientes, seria uma mais valia, já que as ferramentas estudadas têm integração para tal, estudar o mesmo comportamento nos ambiente *Unix/Linux*.

Quanto ao cliente descrito anteriormente, como é um projeto de longa duração que se vai prolongar por vários anos, houve vários temas que estão por desenvolver e outros por melhorar. Foi feito um trabalho notório para conhecer, gerir e estabilizar a infraestrutura.

Quanto aos temas estabelecido com o cliente, apesar da arquitetura da solução de monitorização conseguir monitorizar perfeitamente os servidores existem melhoramentos quanto ao dimensionamento do *Management Server*, ou seja, ter mais um *Management Server* para permitir o Failover de um dos servidores sem interromper o serviço de monitorização da infraestrutura. A alarmística causada pelo SCOM ainda é bastante ruidosa e carece junto do cliente uma granulação desses alertas, a sensibilidade será adquirida com o uso da ferramenta.

Quanto as tarefas de manutenção do SCCM provinham já com uma "bagagem"nesta área, apesar de ser uma ferramenta super poderosa e perigosa os objetivos foram superados com uma correta organização dos recursos e constante monitorização do estado deles. Devido ainda a uma quantidade considerável de servidores *Windows 2003 e 2008R2* existirá um trabalho de migração de cerca de 100 servidores, alguns para a nuvem *Azure*, outros para servidores mais recentes.

Neste último cliente, a terceira fase do projeto sobre a automação de tarefas não foi implementada a tempo útil da apresentação da tese apesar de existir a documentação que permite a automação desses processos, estimando-se uma redução substancial do tempo e recursos atribuídos na resolução dos mesmos. Adicionalmente deverá ser possível solucionar alguns alertas provenientes do SCOM com causas já conhecidas de forma automática com a introdução do SCORCH.

Outro ponto interessante seria desenvolver a aplicabilidade do SCORCH com o SCCM. Teoricamente é possível pois na prática executar um *runbook* é executar *PowerShell*. Existindo um módulo SCCM no *PowerShell* é uma questão de estudar a aplicabilidade ou utilidade dos automatismos a introduzir.

Bibliografia

[1] *.NET*. URL: http://looselycoupled.com/glossary/.NET.

[2] *Disciplina Cloud Computing DI FCT UNL - Aula02-2-21set-EN.pdf-Aula07-1-12out-EN.pdf*.

[3] T. G. Peter Mell. "The NIST Definition of Cloud Computing". Em: (2011). URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

[4] Q. Zhang, L. Cheng e R. Boutabaf. "Cloud computing: state-of-the-art and research challenges". Em: (2010). URL: https://u.cs.biu.ac.il/~ariel/download/ds590/resources/cloud/cloud_sota.pdf.

[5] *O que é uma cloud híbrida?* URL: https://azure.microsoft.com/pt-pt/overview/what-is-hybrid-cloud-computing/.

[6] *Which are the big cloud computing companies?* URL: https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-from-public-and-private-cloud-to-software-as-a/.

[7] E. Schouten. *IBM SmartCloud Essentials*. Packt Publishing Ltd., 2013. ISBN: 978-1782170648.

[8] *What is Nagios*. URL: https://searchitoperations.techtarget.com/definition/Nagios.

[9] W. Barth. *Nagios, 2nd Edition: System and Network Monitoring*. No Starch Press, 2008. ISBN: 978-1593271794.

[10] K. Greene. *Getting Started with Microsoft System Center Operations Manager*. Packt Publishing Ltd., 2016. ISBN: 978-1785289743.

[11] *SCOM – Authoring History And System Center Visual Studio Authoring Extensions 2015*. URL: https://www.stefanroth.net/2015/12/10/scom-authoring-history-and-system-center-visual-studio-authoring-extensions-2015/.

[12] *Operations Manager Management Pack Authoring - Classes and Relationships*. URL: https://social.technet.microsoft.com/wiki/contents/articles/14256.operations-manager-management-pack-authoring-classes-and-relationships.aspx.

[13] *Operations Manager Management Pack Authoring - Choosing a Base Class*. URL: https://social.technet.microsoft.com/wiki/contents/articles/14258.operations-manager-management-pack-authoring-choosing-a-base-class.aspx.

[14] *Operations Manager Management Pack Authoring - Unit Monitors*. URL: https://social.technet.microsoft.com/wiki/contents/articles/15207.operations-manager-management-pack-authoring-unit-monitors.aspx.

[15] *Operations Manager Management Pack Authoring - Aggregate and Dependency Monitors*. URL: https://social.technet.microsoft.com/wiki/contents/articles/15209.operations-manager-management-pack-authoring-aggregate-and-dependency-monitors.aspx.

[16] *Operations Manager Management Pack Authoring - Rules*. URL: https://social.technet.microsoft.com/wiki/contents/articles/15213.operations-manager-management-pack-authoring-rules.aspx.

[17] *Operations Manager Management Pack Authoring - Tasks*. URL: https://social.technet.microsoft.com/wiki/contents/articles/15214.operations-manager-management-pack-authoring-tasks.aspx.

[18] *Openstack*. URL: https://www.openstack.org/.

[19] A. Asp, A. Baumgarten, S. Beaumontand, S. Buchanan e D. Gasser. *Microsoft System Center 2016 Service Manager Cookbook - Second Edition*. Packt Publishing Ltd., 2017. ISBN: 978-1786464897.

[20] D. Seaman. *Installing System Center Service Manager 2012 Part 1*. URL: https://www.derekseaman.com/2012/08/installing-system-center-service.html.

[21] *System Center Configuration Manager (SCCM) 2012 R2 Upgrade ... The first stepsr*. URL: http://techrant.nhflorida.com/system-center-configuration-manager-sccm-2012-r2-upgrade-the-first-steps/.

[22] *Site components for Configuration Manager*. URL: https://docs.microsoft.com/en-us/sccm/core/servers/deploy/configure/site-components.

[23] E. Palmer. "OOPS, IT DIDN'T ARM. - A CASE STUDY OF TWO AUTOMATION SURPRISES". Em: (1995). URL: https://www.researchgate.net/publication/247058880_Oops_it_didn%5C%27t_arm_A_case_study_of_two_automation_surprises.

[24] *Ansible (software)*. URL: https://www.ansible.com/integrations/infrastructure.

[25] *Puppet vs. Chef vs. Ansible vs. SaltStack*. URL: https://www.intigua.com/blog/puppet-vs.-chef-vs.-ansible-vs.-saltstack.

[26] *An Overview of Chef*. URL: https://docs.chef.io/chef_overview.html.

[27] *Chef (software)*. URL: https://www.theregister.co.uk/2015/10/07/chef_introduction/.

[28] *PowerShell*. URL: https://docs.microsoft.com/pt-pt/powershell/.

[29] *Windows PowerShell Desired State Configuration Overview*. URL: https://docs.microsoft.com/en-us/powershell/dsc/overview/overview.

[30] *Push and Pull Configuration Modes*. URL: https://devblogs.microsoft.com/powershell/push-and-pull-configuration-modes/.

[31] *Salt (software)*. URL: https://www.saltstack.com/solutions/it-operations/.

[32] *Puppet (software)*. URL: https://www.chef.io/puppet/.

[33] *Infrastructure as code*. URL: https://resources.sei.cmu.edu/asset_files/WhitePaper/2019_019_001_539335.pdf.

[34] S. Erskine, A. Baumgarten e S. Beaumont. *Microsoft System Center 2012 Orchestrator Cookbook*. Packt Publishing Ltd., 2013. ISBN: 978-1-84968-850-5.

[35] *Azure Resource Manager overview*. URL: https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview.

[36] *active-directory*. URL: https://carlwebster.com/downloads/download-info/active-directory-2/.

[37] *Automatically deploy software updates*. URL: https://docs.microsoft.com/en-us/sccm/sum/deploy-use/automatically-deploy-software-updates.

[38] *Background Intelligent Transfer Service*. URL: https://docs.microsoft.com/en-us/windows/win32/bits/background-intelligent-transfer-service-portal.

ANEXO I

SCOM - Dashboards

Com o objetivo de monitorizar o estado da infraestrutura foram criados 4 *dashboards* personalizados que contêm respetivamente a informação sobre o estado dos discos, memória, CPU e rede nas últimas 24 horas. Para efeitos de demonstração só foi disponibilizado o código para a classe "Discos Lógicos", pois as outras só diferem quanto à propriedade e aos contadores de desempenho.

Adequado para ambientes com aproximadamente menos de 50 máquinas.

```
1
2  $class = get-scomclass -Name Microsoft.Windows.Server.LogicalDisk
3  $serverOSes = Get-SCOMClassInstance -class $class
4
5  #//////////////////////////////////
6  #Alternatively, use a group of OS monitored objects instead:
7  #$serverOSes = get-scomgroup -displayname "*Test Custom Group*" |
       Get-SCOMClassInstance
8  #//////////////////////////////////
9
10 $avg_stat = @{}
11 $dataObjects = @()
12
13 $unitReplacements = @{
14  "Size (Bytes) (String)" = @{ "name" = "Size (GB)"; "coeff" =
       1024*1024*1024};
15  "Free Megabytes" = @{ "name" = "Free GB"; "coeff" = 1024};
16  }
17
18 #//////// Functions Section ////////////////////
19 function RecalculateMinMaxForAvgStatItem {
20  param($name, $value)
```

```
21
22  $avg_stat[$name]["min"] = ($avg_stat[$name]["min"], $value | Measure
       -Min).Minimum
23  $avg_stat[$name]["max"] = ($avg_stat[$name]["max"], $value | Measure
       -Max).Maximum
24 }
25
26 # Function to convert results to PerformanceDataStatistics type to
       return the collection to the VMM column generator component of the
       State Widget:
27 function CreateStatistics {
28 param($value)
29
30 $stat = $ScriptContext.CreateInstance("xsd://Microsoft.SystemCenter.
       Visualization.Library!Microsoft.SystemCenter.Visualization.
       DataProvider/PerformanceDataStatistics")
31 if ($value -ne $null) {
32   $stat["AverageValue"] = [double]$value
33   $stat["Value"] = [double]$value
34  }
35       $stat
36  }
37
38 # Initialize Stat Item:
39 function InitAvgStatItem {
40  param($name)
41
42  if ($avg_stat[$name] -eq $null) {
43    $avg_stat[$name] = @{}
44    $avg_stat[$name]["min"] = 0
45    $avg_stat[$name]["max"] = [Int32]::MinValue
46  }
47  }
48
49 function AddColumnValue {
50  param($dataObject, $name, $value)
51  $v = $value
52  # Transform units value
53   if($unitReplacements[$name] -ne $null) {
54    $r = $unitReplacements[$name]
55    if ($v -ne $null)   {
56     $v = $v = $v / $r["coeff"]
57    }
58    $name = $r["name"]
59  }
60  InitAvgStatItem $name
```

```
61    if ($v -ne $null) {
62     $dataObject[$name] = CreateStatistics($v)
63     RecalculateMinMaxForAvgStatItem $name $v
64    }
65    else {
66     $dataObject[$name] = $null
67    }
68   }
69
70  #///////// Main Section ////////////////////
71  foreach ($serverOS in $serverOSes) {
72   $dataObject = $ScriptContext.CreateFromObject($serverOS, "Id=Id,State
       =HealthState,Name=Name", $null)
73
74   $dataObject["Server"]=$serverOS.Path
75
76   if ($dataObject -ne $null)  {
77    #Get values of Logical Disk properties:
78    $properties = @('SizeNumeric')
79    $properties | % {
80     $prop = $serverOS."[Microsoft.Windows.Server.LogicalDisk].Size"
81     AddColumnValue $dataObject $prop.Type.DisplayName $prop.Value
82    }
83
84    #Last 24 hours
85    $aggregationInterval = 24
86    $dt = New-TimeSpan -hour $aggregationInterval
87    $now = Get-Date
88    $from = $now.Subtract($dt)
89
90    $perfRules = $serverOS.GetMonitoringPerformanceData()
91    foreach ($perfRule in $perfRules)     {
92
93     #Get Free Megabytes
94     if($perfRule.CounterName -eq "Free Megabytes")   {
95      $data = $perfRule.GetValues($from, $now) | % { $_.SampleValue } |
       Measure-Object -Average
96      AddColumnValue $dataObject $perfRule.CounterName $data.Average
97     }
98     #Get Current Disk Queue Length
99     if($perfRule.CounterName -eq "Current Disk Queue Length")  {
100     $data = $perfRule.GetValues($from, $now) | % { $_.SampleValue } |
       Measure-Object -Average
101     AddColumnValue $dataObject $perfRule.CounterName $data.Average
102    }
103    #Get Avg. Disk sec/Transfer
```

```
104     if($perfRule.CounterName -eq "Avg. Disk sec/Transfer")  {
105      $data = $perfRule.GetValues($from, $now) | % { $_.SampleValue } |
        Measure-Object -Average
106      AddColumnValue $dataObject $perfRule.CounterName $data.Average
107     }
108     #Get % Free Space
109     if($perfRule.CounterName -eq "% Free Space")  {
110      $data = $perfRule.GetValues($from, $now) | % { $_.SampleValue } |
        Measure-Object -Average
111      AddColumnValue $dataObject $perfRule.CounterName $data.Average
112
113     #Get % Used Space
114      $UsedSpace = "% Used Space"
115      $value = 100 - $data.Average
116      AddColumnValue $dataObject $UsedSpace $value
117     }
118   }
119   $dataObjects += $dataObject
120     }
121 }
122 $TopN=10
123 $ProcessedObjects = $dataObjects | Sort-Object  {$_["% Free Space"]["
        AverageValue"]} -Descending | Select-Object -First $TopN
124
125 foreach ($dataObject in $ProcessedObjects) {
126  foreach ($metric in $avg_stat.Keys) {
127   $stat = $avg_stat[$metric]
128   $dataObject[$metric]["MinimumValue"] = [double]$stat["min"]
129
130   if ($stat["max"] -ne [Int32]::MinValue) {
131    $dataObject[$metric]["MaximumValue"] = [double]$stat["max"]
132   }
133   else {
134    $dataObject[$metric]["MaximumValue"] = [double]0
135   }
136  }
137  $ScriptContext.ReturnCollection.Add($dataObject)
138 }
```

Para ambientes de grandes dimensões o volume de dados gerado numa consulta à base de dados, com uma taxa de refrescamento do *script* com cerca de um minuto, é inexequível a exibição dos dados, sendo necessária uma abordagem diferente, mais eficiente, composta por duas partes:

1. Criação de uma "Scheduled Task"para executar, de 30 em 30 minutos, o seguinte *script PowerShell* que recolhe as métricas sobre os *Logical Disks* e guarda-os num

ficheiro *.csv*:

```
$class = get-scomclass -Name Microsoft.Windows.Server.LogicalDisk
$serverOSes = Get-SCOMClassInstance -class $class
$object = @()

#Last 24 hours
$aggregationInterval = 24
$dt = New-TimeSpan -hour $aggregationInterval
$now = Get-Date
$from = $now.Subtract($dt)

foreach ($serverOS in $serverOSes) {
  $prop = $serverOS."[Microsoft.Windows.Server.LogicalDisk].Size"
  $perfRules = $serverOS.GetMonitoringPerformanceData()
  foreach ($perfRule in $perfRules)     {
    $data= $perfRule.GetValues($from, $now) | % { $_.SampleValue }
    | Measure-Object -Average
    $object += New-Object -TypeName psobject -Property @{Server=
    $serverOS.Path; Instance=$serverOS; Property=$prop.Type.
    DisplayName; Value=$prop.Value; CounterName=$perfRule.
    CounterName; CounterValue=$data.Average}
  }
}
$object | export-csv -Path c:\DiskData.csv -NoTypeInformation
```

2. Criação de um *PowerShell Grid Widget*, com o seguinte *script PowerShell* que lê os dados do ficheiro *.csv* e constrói as tabelas com o sumário dos 10 servidores com os indicadores mais graves:

```

$file = Import-Csv -Path "c:\DiskData.csv"
$dataObject = $ScriptContext.CreateInstance("xsd://foo!bar/baz")
$var=0
for ($i=0; $i -lt $file.length; $i++) {
    $server = $file[$i].Server
  $instance = $file[$i].Instance
  $prop = $file[$i].Property
  $value = $file[$i].Value
  $cn = $file[$i].CounterName
  $cv = $file[$i].CounterValue

    $dataObject["Id"] = "$var"
  $dataObject["Server"]=$server
  $dataObject["Instance"]=$instance
  $dataObject["Size (GBytes)"]=[math]::Round($value
    /(1024*1024*1024),3)
```

```
17
18
19      if($cn -eq "% Free Space" ){
20      $dataObject["% Free Space"]=[math]::Round($cv,2)
21    }
22    if($cn -eq "Free Megabytes" ){
23      $dataObject["Free Gigabytes"]=[math]::Round($cv/1024,2)
24    }
25      if($cn -eq "Current Disk Queue Length" ){
26      $dataObject["Current Disk Queue Length"]=$cv
27    }
28    if($cn -eq "Avg. Disk sec/Transfer" ){
29      $dataObject["Avg. Disk sec/Transfer"]=[math]::Round($cv,4)
30    }
31    if($cn -eq "% Idle Time" ){
32      $dataObject["% Idle Time"]=[math]::Round($cv,2)
33    }
34
35
36    if(0 -eq ($i-4) % 5){
37          $var =$i
38       $ScriptContext.ReturnCollection.Add($dataObject)
39          $dataObject = $null
40          $dataObject = $ScriptContext.CreateInstance("xsd://foo!bar/
      baz")
41    }
42  }
43
```

ANEXO II

SCCM - SQL Daily Task Example

```
1  Select
2  (
3  Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
4  select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%')
5  )as 'Total',
6  (
7  Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
8  select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1
9  )as 'Assigned',
10 (
11 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
12 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned != 1
13 )as 'NotAssigned',
14 (
15 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
16 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1
17 )as 'Installed',
18 (
19 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
```

```
20 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient != 1
21 )as 'NotInstalled',
22 (
23 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
24 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsObsolete
        = 0
25 )as 'NonObsolete',
26 (
27 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
28 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsObsolete
        != 0
29 )as 'Obsolete',
30 (
31 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
32 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete = 0
33 )as 'Active',
34 (
35 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
36 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        0 and IsObsolete = 0
37 )as 'InActive',
38 (
39 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
40 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete = 0
41 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,
       GetDate())<23)
42 and ResourceID in (select ResourceID from v_GS_WORKSTATION_STATUS
       where DATEDIFF (day,LastHWScan,GetDate())<23)
43 ) as 'HW<30Days',
44 (
45 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
```

```
46 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
47 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in
48 ('Heartbeat Discovery')and DATEDIFF (day,AgentTime,GetDate())<23)
49 and ResourceID Not in (select ResourceID from v_GS_WORKSTATION_STATUS
       where DATEDIFF (day,LastHWScan,GetDate())<23)
50 ) as 'HW>30Days',
51 (
52 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
53 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
54 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,
       GetDate())<23)
55 and ResourceID in (select ResourceID from v_GS_LastSoftwareScan where
       DATEDIFF (day,LastScanDate,GetDate())<23)
56 ) as 'SW<30Days',
57 (
58 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
59 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
60 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,
       GetDate())<23)
61 and ResourceID Not in (select ResourceID from v_GS_LastSoftwareScan
       where DATEDIFF (day,LastScanDate,GetDate())<23)
62 ) as 'SW>30Days',
63 (
64 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
65 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
66 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,
       GetDate())<23)
67 and ResourceID in (select ResourceID from v_UpdateScanStatus where
       lastErrorCode = 0 and DATEDIFF (day,LastScanTime,GetDate())<23)
68 ) as 'WSUS<30Days',
69 (
```

```
70 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
71 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsObsolete
        != 1 and IsActive = 1
72 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,
       GetDate())<23)
73 and ResourceID Not in (select ResourceID from v_UpdateScanStatus where
        lastErrorCode = 0 and DATEDIFF (day,LastScanTime,GetDate())<23)
74 ) as 'WSUS>30Days',
75 (
76 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
77 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
78 and (ResourceID in (select ResourceID from v_GS_WORKSTATION_STATUS
       where DATEDIFF (day,LastHWScan,GetDate())<23)
79 and ResourceID in (select ResourceID from v_GS_LastSoftwareScan where
       DATEDIFF (day,LastScanDate,GetDate())<23))
80 ) as 'Healthy',
81 (
82 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
83 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%')
84 and (ResourceID Not in (select ResourceID from v_GS_WORKSTATION_STATUS
        where DATEDIFF (day,LastHWScan,GetDate())<23)
85 or ResourceID Not in (select ResourceID from v_GS_LastSoftwareScan
       where DATEDIFF (day,LastScanDate,GetDate())<23))
86 ) as 'UnHealthy',
87 cast( (Select((
88 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
89 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
90 and (ResourceID in (select ResourceID from v_GS_WORKSTATION_STATUS
       where DATEDIFF (day,LastHWScan,GetDate())<23)
91 and ResourceID in (select ResourceID from v_GS_LastSoftwareScan where
       DATEDIFF (day,LastScanDate,GetDate())<23))
92 )/(
93 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
```

```
94 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1
95 ))*100) as decimal(5,2))as 'Healthy%',
96 cast( (Select((
97 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
98 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
99 and ResourceID in (select ResourceID from v_AgentDiscoveries Where
       AgentName in
100 ('Heartbeat Discovery') and DATEDIFF (day,AgentTime,GetDate())<23)
101 and ResourceID in (select ResourceID from v_UpdateScanStatus where
       lastErrorCode = 0 and DATEDIFF (day,LastScanTime,GetDate())<23)
102 )/(
103 Select COUNT(distinct(Name)) from v_FullCollectionMembership where
       ResourceID in (
104 select ResourceID from v_R_System where Operating_System_Name_and0
       like '%Server%') and IsAssigned = 1 and IsClient = 1 and IsActive =
        1 and IsObsolete != 1
105 and (ResourceID in (select ResourceID from v_GS_WORKSTATION_STATUS
       where DATEDIFF (day,LastHWScan,GetDate())<23)
106 and ResourceID in (select ResourceID from v_GS_LastSoftwareScan where
       DATEDIFF (day,LastScanDate,GetDate())<23))
107 ))*100) as decimal(5,2))as 'WSUS%'
```

Com o seguinte resultado:

Tabela II.1: *Servers Agent Health Status.*

Total	406
Assigned	406
NotAssigned	0
Installed	311
NotInstalled	95
NonObsolete	311
Obsolete	0
Active	311
InActive	0
HW<30Days	291
HW>30Days	2
SW<30Days	292
SW>30Days	1
WSUS<30Days	278
WSUS>30Days	15
Healthy	291
UnHealthy	115

ANEXO III

SCORCH - Auto-fecho de Pedidos de Serviço

```
# Get List of all Service Request modified in the last 2 days with an
    InProgress status
$ServiceRequestClass = Get-SCSMClass -Name System.WorkItem.
    ServiceRequest$
$ServiceRequestCompletedStatus = Get-SCSMEnumeration -Name
    ServiceRequestStatusEnum.Completed$
$ServiceRequestCompletedStatusID = $ServiceRequestCompletedStatus.ID
$TwoDaysAgo = $((Get-Date).AddDays(-2))

Get-SCSMObject -Class $ServiceRequestClass -Filter "CompletedDate &lt;
    '$TwoDaysAgo' AND Status = '$ServiceRequestCompletedStatusID'" |
    Set-SCSMObject -Property Status -Value "Closed"
```